

Trabajo Fin de Máster
Máster Universitario en Ingeniería Industrial

**Análisis comparativo de la asignación de
cabinas a llamadas en transporte vertical
de arquitectura Double Deck mediante
metaheurísticas.**

Autor: Alejandro Vázquez Ledesma

Tutor: Pablo Fabio Cortés Achedad

Dep. Organización Industrial y Gestión de Empresas II
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2018



Trabajo Fin de Máster
Máster Universitario en Ingeniería Industrial

Análisis comparativo de la asignación de cabinas a llamadas en transporte vertical de arquitectura Double Deck mediante metaheurísticas

Autor:

Alejandro Vázquez Ledesma

Tutor:

Pablo Fabio Cortés Achedad

Catedrático de Universidad

Dep. Organización y Gestión de Empresas II

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2018

Trabajo Fin de Máster: Análisis comparativo de la asignación de cabinas a llamadas en transporte vertical de arquitectura Double Deck mediante metaheurísticas

Autor: Alejandro Vázquez Ledesma

Tutor: Pablo Fabio Cortés Achedad

El tribunal nombrado para juzgar el TFM arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2018

El Secretario del Tribunal

Agradecimientos

A Pablo Cortés, por su confianza depositada para trabajar en el departamento y realizar este Trabajo Fin de Máster, así como por la ayuda y atención prestada y su celeridad a la hora de responder cualquier tipo de duda o problema.

A mi pareja y familia, por su paciencia infinita, confianza, ánimo y apoyo a lo largo del camino a modo de ayuda para que todos los objetivos y logros marcados fuesen cumplidos de forma satisfactoria.

A los que ya no están pero que aportaron su granito de arena.

Muchas Gracias

El encarecimiento del precio del metro cuadrado de terreno en las grandes ciudades hace que la construcción vertical presente un continuo desarrollo y que el número de edificios altos y rascacielos se encuentre en auge. Debido a la gran cantidad de personas que hay que transportar en el interior del edificio se hace necesaria la instalación de grupos de ascensores eficientes que logren la minimización del tiempo medio de espera de los pasajeros frente a la puerta del ascensor, así como la minimización del consumo energético del edificio. Este Trabajo Fin de Máster realiza un análisis comparativo, para edificios con ascensores de arquitectura *Double Deck*, de tiempos (tiempo medio de espera, tránsito y total) y del consumo energético de las cabinas para tres patrones de tráfico diferentes (interfloor, lunchpeak y downpeak). Para cada uno de los patrones se tienen en cuenta dos funciones de evaluación diferentes (basada en tiempo y basada en distancia) y tres métodos de resolución (búsqueda tabú, algoritmo genético y DC-Elevate) para la asignación de cabinas a llamadas.

Abstract

The increase of the price of the floor in the bigger cities have done that vertical construction has a continuous development. Moreover, the number of high-rise buildings and people who work in the building are raising, so it is necessary to develop very efficient group of elevators. This group of elevators must minimize the average waiting time and the energetic consumption of the cabins. This work carry out a comparison of times (average waiting time, average journey time and average total time) and consumption for buildings with *Double Deck* elevators and three different traffic patterns (interfloor, lunchpeak and downpeak). In addition, it is used two evaluation functions (based in time and based in distance) and three resolution ways (tabu search, genetic algorithm and DC-Elevate) for allocating the cabins to calls of passengers.

AGRADECIMIENTOS.....	- 7 -
RESUMEN.....	- 9 -
ABSTRACT	- 10 -
ÍNDICE DE FIGURAS.....	- 13 -
ÍNDICE DE TABLAS.....	- 17 -
1. OBJETO	- 18 -
2. INTRODUCCIÓN A LOS SISTEMAS DE ASCENSORES	- 20 -
2.1. RESEÑA HISTÓRICA.....	- 20 -
2.2. TIPOS DE ASCENSORES	- 24 -
2.2.1. <i>Según su aspecto técnico</i>	- 24 -
2.2.2. <i>Según su aspecto de diseño</i>	- 27 -
2.3. FUNCIONAMIENTO DEL ASCENSOR	- 33 -
3. TRANSPORTE VERTICAL.....	- 35 -
3.1. PATRONES DE TRÁFICO	- 35 -
3.2. NIVEL DE SERVICIO	- 37 -
4. ASIGNACIÓN DE CABINAS A LLAMADAS	- 39 -
4.1. DEFINICIÓN DEL PROBLEMA	- 39 -
i. <i>Restricciones generales</i>	- 39 -
ii. <i>Restricciones particulares</i>	- 40 -
4.2. MÉTODOS DE ASIGNACIÓN	- 40 -
4.2.1. <i>Asignación basada en distancia</i>	- 40 -
4.2.2. <i>Asignación basada en tiempo</i>	- 42 -
4.3. MÉTODOS DE RESOLUCIÓN.....	- 43 -
4.3.1. <i>Búsqueda Tabú</i>	- 44 -
4.3.2. <i>Algoritmo Genético</i>	- 48 -
4.3.3. <i>Destination Control Elevate</i>	- 51 -
5. EXPERIMENTACIÓN	- 52 -
5.1. CARACTERÍSTICAS DE LA EXPERIMENTACIÓN	- 52 -
5.1.1. <i>Edificios</i>	- 52 -
5.1.2. <i>Patrones de tráfico</i>	- 53 -
5.1.3. <i>Ascensores</i>	- 53 -

5.2. RESULTADOS DE LA EXPERIMENTACIÓN	- 54 -
5.2.1. Tráfico Interfloor.....	- 56 -
5.2.2. Tráfico Downpeak.....	- 58 -
5.2.3. Tráfico Lunchpeak.....	- 60 -
6. ANÁLISIS COMPARATIVO DE TIEMPOS	- 62 -
6.1. FUNCIÓN DE EVALUACIÓN BASADA EN DISTANCIA	- 62 -
6.1.1. Tráfico Interfloor.....	- 62 -
6.1.2. Tráfico Lunchpeak.....	- 73 -
6.1.3. Tráfico Downpeak.....	- 83 -
6.2. FUNCIÓN DE EVALUACIÓN BASADA EN TIEMPO.....	- 94 -
6.2.1. Tráfico Interfloor.....	- 94 -
6.2.2. Tráfico Lunchpeak.....	- 104 -
6.2.3. Tráfico Downpeak.....	- 115 -
7. ANÁLISIS COMPARATIVO DE ENERGÍA	- 126 -
7.1. TRÁFICO INTERFLOOR	- 126 -
7.2. TRÁFICO LUNCHPEAK	- 127 -
7.3. TRÁFICO DOWNPEAK.....	- 128 -
8. ALGORITMOS MÁS EFICIENTES SEGÚN LAS CARACTERÍSTICAS DEL EDIFICIO	- 130 -
8.1. TRÁFICO INTERFLOOR	- 130 -
8.2. TRÁFICO LUNCHPEAK	- 134 -
8.3. TRÁFICO DOWNPEAK.....	- 138 -
8.4. PROPUESTA DE APLICACIÓN DE PARÁMETROS DE DESEMPEÑO.....	- 141 -
9. CONCLUSIONES	- 154 -
10. REFERENCIAS.....	- 156 -
ANEXO A: SOFTWARE	- 158 -
1.1. ELEVATE 8	- 158 -
1.1.1. Características del ascensor y edificio	- 159 -
1.2. MICROSOFT VISUAL STUDIO 2008.....	- 165 -
1.2.1. Interfaz de desarrollador.....	- 165 -
1.2.2. Desarrollo en Microsoft Visual Studio.....	- 166 -
ANEXO B: GRÁFICAS EXPERIMENTALES.....	- 167 -
ANEXO C: VARIABLES DE OBJETO	- 191 -

Índice de figuras

Figura 1: Elisha Graves muestra su “freno de emergencia” (Wikimedia Commons)	- 20 -
Figura 2: Logo Mitsubishi Electric	- 22 -
Figura 3: Logo Otis	- 22 -
Figura 4: Logo Schindler	- 22 -
Figura 5: Logo Thyssenkrupp	- 23 -
Figura 6: Logo MP	- 23 -
Figura 7: Logo Orona	- 23 -
Figura 8: Ascensor autoportante (Wikimedia Commons)	- 24 -
Figura 9: Ascensor hidráulico (Wikimedia Commons)	- 25 -
Figura 10: Ascensor electromecánico (Wikimedia Commons)	- 26 -
Figura 11: Ascensor panorámico (Wikimedia Commons)	- 27 -
Figura 12: Sistema de ascensores multicabina (TyssenKrupp)	- 28 -
Figura 13: Ascensor Double Deck (Wikimedia Commons)	- 29 -
Figura 14: Sistema de ascensores Double Deck (Wikimedia Commons)	- 29 -
Figura 15: Stratosphere, Las Vegas (Wikimedia Commons)	- 30 -
Figura 16: Torre Picasso, Madrid (Wikimedia Commons)	- 31 -
Figura 17: The Shard (izquierda) junto a Tower Bridge, Londres (Wikimedia Commons)	- 32 -
Figura 18: Citigroup Center, New York (Wikimedia Commons)	- 32 -
Figura 19: Torre Burj Khalifa, Dubai (Wikimedia Commons)	- 33 -
Figura 20: Mecanismo de contrapeso de un ascensor (Wikimedia Commons)	- 34 -
Figura 21: Patrón de tráfico Siikonen (Cortés et al., 2009)	- 35 -
Figura 22: Asignación de llamadas a cabinas basada en distancia	- 41 -
Figura 23: Asignación de cabinas a llamadas basada en tiempo	- 42 -
Figura 24: Diagrama de flujo Búsqueda Tabú	- 45 -
Figura 25: Asignación de llamadas a ascensores	- 48 -
Figura 26: Cruce de individuos	- 49 -
Figura 27: Mutación de individuos	- 50 -
Figura 28: Ejemplo gráfica AWT	- 54 -
Figura 29: Ejemplo gráfica AJT	- 54 -
Figura 30: Ejemplo consumo energético	- 55 -
Figura 31: AWT (función de evaluación: distancia)- 12 Plantas. Interfloor	- 62 -
Figura 32: AJT (función de evaluación: distancia)- 12 Plantas. Interfloor	- 63 -
Figura 33: ATT (función de evaluación: distancia)- 12 Plantas. Interfloor	- 63 -
Figura 34: AWT (función de evaluación: distancia)- 20 Plantas. Interfloor	- 64 -
Figura 35: AJT (función de evaluación: distancia)- 20 Plantas. Interfloor	- 64 -
Figura 36: ATT (función de evaluación: distancia)- 20 Plantas. Interfloor	- 65 -
Figura 37: AWT (función de evaluación: distancia)- 24 Plantas. Interfloor	- 65 -
Figura 38: AJT (función de evaluación: distancia)- 24 Plantas. Interfloor	- 66 -
Figura 39: ATT (función de evaluación: distancia)- 24 Plantas. Interfloor	- 66 -
Figura 40: AWT (función de evaluación: distancia)- 28 Plantas. Interfloor	- 67 -
Figura 41: AJT (función de evaluación: distancia)- 28 Plantas. Interfloor	- 67 -
Figura 42: ATT (función de evaluación: distancia)- 28 Plantas. Interfloor	- 68 -
Figura 43: AWT (función de evaluación: distancia)- 32 Plantas. Interfloor	- 68 -
Figura 44: AJT (función de evaluación: distancia)- 32 Plantas. Interfloor	- 69 -
Figura 45: ATT (función de evaluación: distancia)- 32 Plantas. Interfloor	- 69 -
Figura 46: AWT (función de evaluación: distancia)- 36 Plantas. Interfloor	- 70 -
Figura 47: AJT (función de evaluación: distancia)- 36 Plantas. Interfloor	- 70 -

Figura 103: AWT (función de evaluación: tiempo)- 28 Plantas. Interfloor	- 98 -
Figura 104: AJT (función de evaluación: tiempo)- 28 Plantas. Interfloor	- 99 -
Figura 105: ATT (función de evaluación: tiempo)- 28 Plantas. Interfloor	- 99 -
Figura 106: AWT (función de evaluación: tiempo)- 32 Plantas. Interfloor	- 100 -
Figura 107: AJT (función de evaluación: tiempo)- 32 Plantas. Interfloor	- 100 -
Figura 108: ATT (función de evaluación: tiempo)- 32 Plantas. Interfloor	- 101 -
Figura 109: AWT (función de evaluación: tiempo)- 36 Plantas. Interfloor	- 101 -
Figura 110: AJT (función de evaluación: tiempo)- 36 Plantas. Interfloor	- 102 -
Figura 111: ATT (función de evaluación: tiempo)- 36 Plantas. Interfloor	- 102 -
Figura 112: AWT (función de evaluación: tiempo)- 40 Plantas. Interfloor	- 103 -
Figura 113: AJT (función de evaluación: tiempo)- 40 Plantas. Interfloor	- 103 -
Figura 114: ATT (función de evaluación: tiempo)- 40 Plantas. Interfloor	- 104 -
Figura 115: AWT (función de evaluación: tiempo)- 12 Plantas. Lunchpeak	- 104 -
Figura 116: AJT (función de evaluación: tiempo)- 12 Plantas. Lunchpeak	- 105 -
Figura 117: ATT (función de evaluación: tiempo)- 12 Plantas. Lunchpeak	- 105 -
Figura 118: AWT (función de evaluación: tiempo)- 20 Plantas. Lunchpeak	- 106 -
Figura 119: AJT (función de evaluación: tiempo)- 20 Plantas. Lunchpeak	- 106 -
Figura 120: ATT (función de evaluación: tiempo)- 20 Plantas. Lunchpeak	- 107 -
Figura 121: AWT (función de evaluación: tiempo)- 24 Plantas. Lunchpeak	- 107 -
Figura 122: AJT (función de evaluación: tiempo)- 24 Plantas. Lunchpeak	- 108 -
Figura 123: ATT (función de evaluación: tiempo)- 24 Plantas. Lunchpeak	- 108 -
Figura 124: AWT (función de evaluación: tiempo)- 28 Plantas. Lunchpeak	- 109 -
Figura 125: AJT (función de evaluación: tiempo)- 28 Plantas. Lunchpeak	- 109 -
Figura 126: ATT (función de evaluación: tiempo)- 28 Plantas. Lunchpeak	- 110 -
Figura 127: AWT (función de evaluación: tiempo)- 32 Plantas. Lunchpeak	- 110 -
Figura 128: AJT (función de evaluación: tiempo)- 32 Plantas. Lunchpeak	- 111 -
Figura 129: ATT (función de evaluación: tiempo)- 32 Plantas. Lunchpeak	- 111 -
Figura 130: AWT (función de evaluación: tiempo)- 36 Plantas. Lunchpeak	- 112 -
Figura 131: AJT (función de evaluación: tiempo)- 36 Plantas. Lunchpeak	- 112 -
Figura 132: ATT (función de evaluación: tiempo)- 36 Plantas. Lunchpeak	- 113 -
Figura 133: AWT (función de evaluación: tiempo)- 40 Plantas. Lunchpeak	- 113 -
Figura 134: AJT (función de evaluación: tiempo)- 40 Plantas. Lunchpeak	- 114 -
Figura 135: ATT (función de evaluación: tiempo)- 40 Plantas. Lunchpeak	- 114 -
Figura 136: AWT (función de evaluación: tiempo)- 12 Plantas. Downpeak	- 115 -
Figura 137: AJT (función de evaluación: tiempo)- 12 Plantas. Downpeak	- 115 -
Figura 138: ATT (función de evaluación: tiempo)- 12 Plantas. Downpeak	- 116 -
Figura 139: AWT (función de evaluación: tiempo)- 20 Plantas. Downpeak	- 116 -
Figura 140: AJT (función de evaluación: tiempo)- 20 Plantas. Downpeak	- 117 -
Figura 141: ATT (función de evaluación: tiempo)- 20 Plantas. Downpeak	- 117 -
Figura 142: AWT (función de evaluación: tiempo)- 24 Plantas. Downpeak	- 118 -
Figura 143: AJT (función de evaluación: tiempo)- 24 Plantas. Downpeak	- 118 -
Figura 144: ATT (función de evaluación: tiempo)- 24 Plantas. Downpeak	- 119 -
Figura 145: AWT (función de evaluación: tiempo)- 28 Plantas. Downpeak	- 119 -
Figura 146: AJT (función de evaluación: tiempo)- 28 Plantas. Downpeak	- 120 -
Figura 147: ATT (función de evaluación: tiempo)- 28 Plantas. Downpeak	- 120 -
Figura 148: AWT (función de evaluación: tiempo)- 32 Plantas. Downpeak	- 121 -
Figura 149: AJT (función de evaluación: tiempo)- 32 Plantas. Downpeak	- 121 -
Figura 150: ATT (función de evaluación: tiempo)- 32 Plantas. Downpeak	- 122 -
Figura 151: AWT (función de evaluación: tiempo)- 36 Plantas. Downpeak	- 122 -
Figura 152: AJT (función de evaluación: tiempo)- 36 Plantas. Downpeak	- 123 -
Figura 153: ATT (función de evaluación: tiempo)- 36 Plantas. Downpeak	- 123 -
Figura 154: AWT (función de evaluación: tiempo)- 40 Plantas. Downpeak	- 124 -
Figura 155: AJT (función de evaluación: tiempo)- 40 Plantas. Downpeak	- 124 -
Figura 156: ATT (función de evaluación: tiempo)- 40 Plantas. Downpeak	- 125 -
Figura 157: Consumo energético (función de evaluación: distancia). Interfloor	- 126 -

Figura 158: Consumo energético (función de evaluación: tiempo). Interfloor	- 127 -
Figura 159: Consumo energético (función de evaluación: distancia). Lunchpeak	- 127 -
Figura 160: Consumo energético (función de evaluación: tiempo). Lunchpeak	- 128 -
Figura 161: Consumo energético (función de evaluación: distancia). Downpeak	- 128 -
Figura 162: Consumo energético (función de evaluación: tiempo). Downpeak	- 129 -
Figura 163: Gráfica 3D N° Ascensores-N° Plantas-AWT (función de evaluación: distancia). Interfloor	- 130 -
Figura 164: Gráfica 3D N° Ascensores-N° Plantas-AWT (función de evaluación: tiempo). Interfloor	- 131 -
Figura 165: Gráfica 3D N° Ascensores-N° Plantas-AJT (función de evaluación: distancia). Interfloor	- 132 -
Figura 166: Gráfica 3D N° Ascensores-N° Plantas-AJT (función de evaluación: tiempo). Interfloor	- 133 -
Figura 167: Gráfica 3D N° Ascensores-N° Plantas-AWT (función de evaluación: distancia). Lunchpeak	- 134 -
Figura 168: Gráfica 3D N° Ascensores-N° Plantas-AWT (función de evaluación: tiempo). Lunchpeak	- 135 -
Figura 169: Gráfica 3D N° Ascensores-N° Plantas-AJT (función de evaluación: distancia). Lunchpeak	- 136 -
Figura 170: Gráfica 3D N° Ascensores-N° Plantas-AJT (función de evaluación: tiempo). Lunchpeak	- 137 -
Figura 171: Gráfica 3D N° Ascensores-N° Plantas-AWT (función de evaluación: distancia). Downpeak	- 138 -
Figura 172: Gráfica 3D N° Ascensores-N° Plantas-AWT (función de evaluación: tiempo). Downpeak	- 139 -
Figura 173: Gráfica 3D N° Ascensores-N° Plantas-AJT (función de evaluación: distancia). Downpeak	- 140 -
Figura 174: Gráfica 3D N° Ascensores-N° Plantas-AJT (función de evaluación: tiempo). Downpeak	- 141 -
Figura 175: Ventana principal Elevate	- 158 -
Figura 176: Ventana Analysis Data	- 159 -
Figura 177: Ventana Building Data	- 160 -
Figura 178: Ventana standard Elevator Data	- 160 -
Figura 179: Ventana advanced Elevator Data	- 161 -
Figura 180: Ventana standard Passenger Data	- 161 -
Figura 181: Ventana advanced Passenger Data	- 162 -
Figura 182: Patrón de tráfico CIBSE	- 162 -
Figura 183: Patrón de tráfico Strackosh	- 163 -
Figura 184: Patrón de tráfico Siikonen	- 163 -
Figura 185: Ventana Report Options	- 164 -
Figura 186: Ventana de simulación de Elevate	- 164 -
Figura 187: Ventana de propiedades de Microsoft Visual Studio	- 165 -
Figura 188: Extracto del algoritmo programado en Visual Studio	- 166 -

Índice de tablas

Tabla 1: Comparativa de carga y velocidad máxima en el tiempo	- 21 -
Tabla 2: Nivel de servicio según porcentaje de llamadas respondidas	- 37 -
Tabla 3: Nivel de servicio en función del tiempo empleado en responder las llamadas	- 37 -
Tabla 4: Tiempos asociados al cálculo de la asignación basada en tiempo	- 43 -
Tabla 5: Características del edificio	- 52 -
Tabla 6: Características de los ascensores	- 53 -
Tabla 7: Tiempos medios de espera, tránsito y total para función de evaluación de distancia. Interfloor	- 56 -
Tabla 8: Tiempos medios de espera, tránsito y total para función de evaluación de tiempo. Interfloor	- 56 -
Tabla 9: Consumo energético para función de evaluación de distancia. Interfloor	- 57 -
Tabla 10: Consumo energético para función de evaluación de tiempo. Interfloor	- 57 -
Tabla 11: Tiempos medios de espera, tránsito y total para función de evaluación de distancia. Downpeak	- 58 -
Tabla 12: Tiempos medios de espera, tránsito y total para función de evaluación de tiempo. Downpeak	- 58 -
Tabla 13: Consumo energético para función de evaluación de distancia. Downpeak	- 59 -
Tabla 14: Consumo energético para función de evaluación de tiempo. Downpeak	- 59 -
Tabla 15: Tiempos medios de espera, tránsito y total para función de evaluación de distancia. Lunchpeak	- 60 -
Tabla 16: Tiempos medios de espera, tránsito y total para función de evaluación de tiempo. Lunchpeak	- 60 -
Tabla 17: Consumo energético para función de evaluación de distancia. Lunchpeak	- 61 -
Tabla 18: Consumo energético para función de evaluación de tiempo. Lunchpeak	- 61 -
Tabla 19: Propuesta de aplicación Interfloor	- 142 -
Tabla 20: Propuesta de aplicación Lunchpeak	- 144 -
Tabla 21: Propuesta de aplicación Downpeak	- 146 -
Tabla 22: Comparativa resolución-evaluación para Interfloor	- 148 -
Tabla 23: Comparativa resolución-evaluación para Lunchpeak	- 150 -
Tabla 24: Comparativa resolución-evaluación para Downpeak	- 152 -

1. OBJETO

Este Trabajo Fin de Máster tiene como objetivo el análisis de la asignación de llamadas a cabinas con arquitectura *Double Deck* mediante el estudio de la resolución del problema con metaheurísticas (Búsqueda Tabú y Algoritmo Genético) y un algoritmo resolutivo del software de simulación (DC-Elevate). Los algoritmos se encuentran implementados en C++ con el software *Microsoft Visual Studio* para su posterior simulación mediante el software *Elevate 8*.

Se realiza un análisis comparativo del tiempo medio de espera, tránsito y total de los pasajeros, así como del consumo energético del motor del ascensor para cada uno de los tres tráficos que se definen (interfloor, downpeak y lunchpeak). Además, se toman para el análisis dos funciones de evaluación diferentes y tres métodos de resolución.

La memoria de este trabajo se divide en seis secciones. Al final se encuentran los anexos con las gráficas generadas por *Elevate* que se obtienen a lo largo del estudio y los códigos de edificios y ascensores que se usan a la hora de implementar en C++ el algoritmo. Además, se incorpora un anexo descriptivo de las herramientas software que se han empleado en el análisis.

En primer lugar, en la segunda sección se realiza una introducción histórica del ascensor y se describe el funcionamiento de este. Después se realiza una organización de los distintos tipos de ascensores existentes poniéndose especial énfasis en las cabinas de tipo *Double Deck*, objeto de este Trabajo Fin de Máster.

En la tercera sección se presentan los diferentes patrones de tráfico existentes en el transporte vertical de un edificio. Se establece en este apartado la medición que se va a realizar para evaluar el nivel de servicio ofrecido para cada edificio en función del método de resolución.

La sección cuarta de este trabajo recoge la descripción del problema de asignación de llamadas a cabinas mediante las restricciones y función objetivo. Se definen, además, las dos diferentes funciones de evaluación que se emplean y los tres métodos de resolución del problema.

La sección quinta recoge las características de la experimentación realizada mediante el software *Elevate 8* definiendo las diferentes combinaciones de número de plantas y número de cabinas de cada edificio y obteniéndose los tiempos medios de espera, tránsito y total, así como el consumo energético.

En la sección sexta se elabora un análisis comparativo de los resultados que se han obtenido en la sección anterior. Concretamente, se realiza un estudio comparativo para cada tráfico en función de los tiempos obtenidos en las simulaciones.

La séptima sección analiza los resultados obtenidos de la experimentación a modo de comparación, para cada tráfico, en función de la energía consumida por los ascensores del edificio.

En la octava sección se muestran los resultados de los algoritmos más eficientes según las características del edificio. Se propone una recomendación de número de cabinas a implementar en cada edificio, así como la función de evaluación y método de resolución que aportan una mayor eficiencia.

En último lugar, se realiza una reseña a modo de resumen sobre la problemática tratada en el documento y se extraen las conclusiones a las que se llega tras el análisis comparativo de los parámetros de desempeño para la asignación de llamadas a cabinas *Double-Deck*.

El primer anexo recoge la descripción de los softwares empleados en el análisis. Además, se explica el modo de proceder para el empleo de la interfaz de desarrollador necesaria para implementar las metaheurísticas programadas en el simulador de edificios y ascensores.

Se tiene un segundo anexo en el cual se recogen las gráficas generadas por el software *Elevate 8* para cada una de las simulaciones realizadas.

En el tercer anexo se incluyen los códigos de los objetos empleados (edificios, ascensores y pasajeros) para la implementación del algoritmo en C++.

2. INTRODUCCIÓN A LOS SISTEMAS DE ASCENSORES

En esta sección segunda se expone la cronología del ascensor en forma de reseña histórica y se describe el funcionamiento que lleva a cabo el ascensor. Además, se presenta una agrupación con los diferentes tipos de ascensores existentes.

2.1. Reseña histórica

En el año 236 d.C. Arquímedes muestra un dispositivo montacargas mediante el cual a través de una cuerda y una polea conseguía levantar una plataforma. Es este hecho el primer documento registrado en la historia del ascensor.

En 1853, Elisha Graves Otis expone un ascensor con “freno de emergencia” que evita la caída libre de la cabina en caso de rotura del cable que la sostiene. Se mostró la eficacia de su sistema de emergencia instalando este tipo de ascensor en el edificio Crystal Palace de New York. En la

Figura 1 se muestra el dispositivo de seguridad (Un poco de historia, 2018).

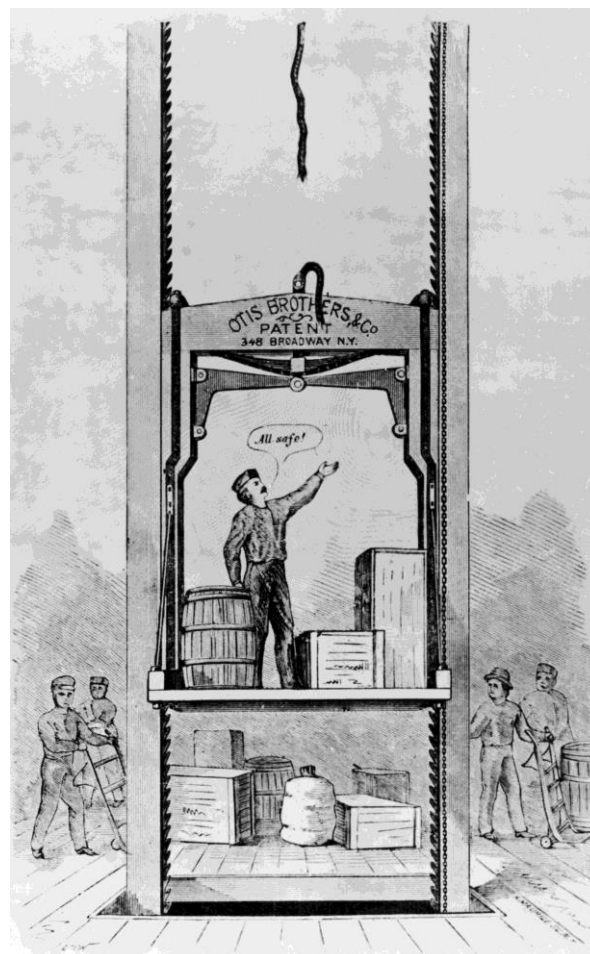


Figura 1: Elisha Graves muestra su “freno de emergencia” (Wikimedia Commons)

En la misma fecha, en Inglaterra, Frost y Stutt desarrollan un ascensor con contrapeso que se acciona mediante tracción cuya denominación fue “*Teagle*” (aparejo para elevación).

En el año 1857, se instala el primer ascensor de pasajeros del mundo, fabricado por *Otis Elevator Company*, en un hotel de cinco pisos en Broadway. Supone este hito un revulsivo importante ya que hasta dicha fecha el alojamiento en altura era inaceptable ya que requería subir demasiadas escaleras con el equipaje. Supone esta fecha un punto de inflexión ya que el número de ascensores comienza a crecer.

En 1867, Leon Edoux presenta en la Exposición de París un ascensor con accionamiento hidráulico. Siemens muestra un ascensor con accionamiento eléctrico en 1880 durante la Exposición de Mannheim, aunque no es hasta 1889 cuando Norton Otis desarrolla el primer ascensor eléctrico accionado mediante corriente continua. En 1900 se introduce el motor de inducción para corriente alterna, siendo en 1903 cuando aparecen en Estados Unidos los ascensores con corriente de tracción sin engranajes.

Se muestra a continuación en la Tabla 1 una comparativa de la carga y velocidad máxima de algunos ascensores a lo largo de la historia, demostrándose cómo la evolución tecnológica con el avance de los años deja patente la mejora en el campo del transporte vertical a modo de conseguir una mayor velocidad y carga máxima.

Tabla 1: Comparativa de carga y velocidad máxima en el tiempo

Año	Carga Máxima (kg)	Velocidad Máxima (m/min)
1857	450	12
1867	-	150
1889	675	30
1922	-	420

Desde el siglo XX hasta hoy, la innovación en los sistemas de ascensores se encuentra en la mejora de los grupos motores, así como en la seguridad que estos ofrecen. Además, la arquitectura de edificios ha cambiado a lo largo de este tiempo haciendo que el requerimiento de ascensores más eficientes sea necesario en forma de ahorro energético y rapidez en el servicio que la cabina lleva a cabo.

Se encuentran diversas empresas que a lo largo de la historia han hecho evolucionar el mundo del ascensor, tales como *Mitsubishi Electrics*, *Otis* o *Schindler*. Aunque también encontramos otra serie de empresas de más reciente creación que favorecen la competitividad del sector en pos de favorecer la evolución del ascensor como *ThyssenKrupp*, *MP* u *Orona*. En las siguientes figuras se muestran los logotipos de las distintas empresas nombradas.



Figura 2: Logo Mitsubishi Electric



Figura 3: Logo Otis



Figura 4: Logo Schindler



Figura 5: Logo Thyssenkrupp



Figura 6: Logo MP



Figura 7: Logo Orona

2.2. Tipos de Ascensores

En este apartado se presenta la tipología de ascensores que se tienen en la actualidad. Se realiza la clasificación de ascensores en función de su aspecto técnico y en función de su aspecto estético de diseño (Tipos de ascensores para las edificaciones, 2018).

2.2.1. Según su aspecto técnico

- **Ascensores autoportantes** (sin sala de máquinas):

Es un tipo de ascensor muy común en lugares como viviendas unifamiliares, salones de fiesta, cines o cualquier otra instalación que deba prescindir de una sala de máquinas debido a la falta de espacio. La máquina de tracción en su conjunto completo se coloca en forma estructural dentro del mismo pasadizo y en su parte superior (cielo de la caja). Se muestra en la Figura 8 este tipo de elevador.

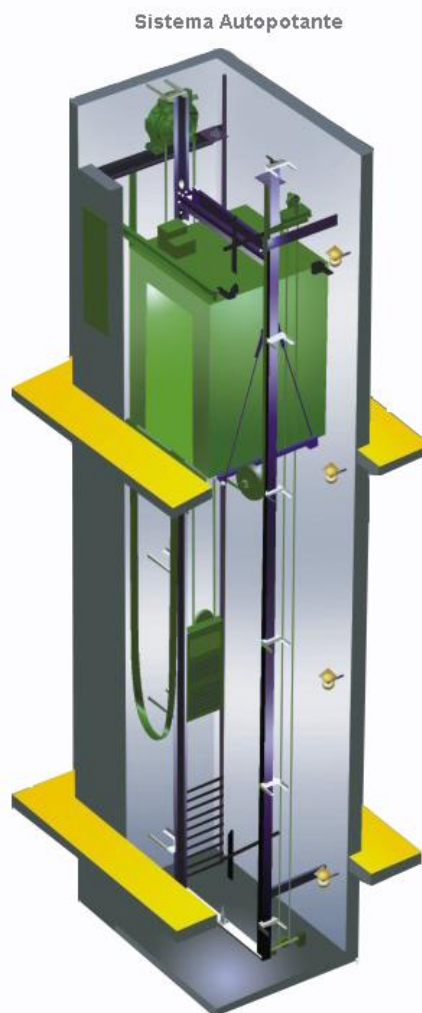


Figura 8: Ascensor autoportante (Wikimedia Commons)

- **Ascensores hidráulicos:**

Se instalan en recorridos cortos para una distancia comprendida entre cuatro y cinco paradas. Su modo de funcionamiento no se realiza a través de máquina de tracción sino que depende de una central oleodinámica que tiene en su interior una bomba sumergida en aceite para controlar el ascenso de la cabina. La sala de máquinas debe estar perfectamente dimensionada y habilitada para la instalación de la central y el control de maniobras por lo que se requiere de un cierto volumen para su ubicación.

La instalación en el pasadizo requiere de un pistón colocado en el centro de una arcatina, que es la que eleva el pistón por medio de una polea de la cual los cables de tracción van unidos desde el punto fijo de la arcatina al punto fijo de la arcata. En la Figura 9 se puede observar este tipo de ascensor.

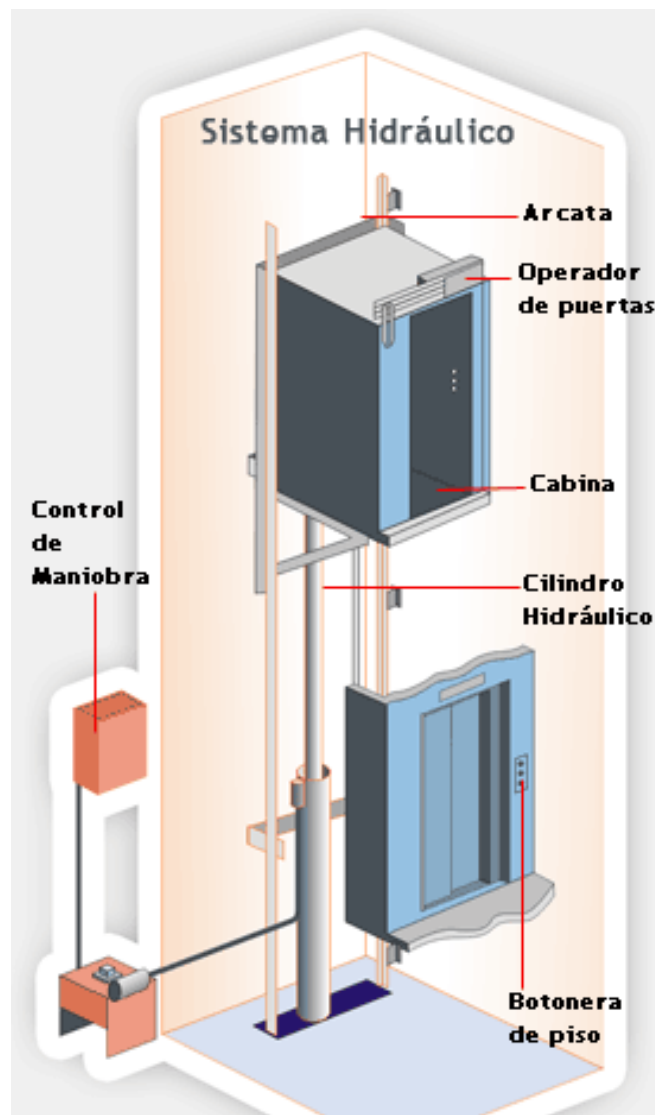


Figura 9: Ascensor hidráulico (Wikimedia Commons)

- **Ascensores electromecánicos:**

Es este el tipo de ascensor más empleado en edificios de viviendas multifamiliares. Este tipo de elevadores, a diferencia del ascensor hidráulico, sí requiere de una máquina de tracción en la sala de máquinas que debe estar ubicada debajo o arriba de la instalación.

La tracción se realiza con motor eléctrico, polea y máquina reductora. De la polea cuelga el cable de tracción que es arrastrado por fricción en su giro, mientras que la cabina es guiada en su trayecto por rieles a lo largo del hueco de la cabina.

Una de las ventajas con las que cuenta este tipo de ascensor es que en caso de que haya una persona encerrada, se puede intervenir manualmente accionando una manivela de freno, cortando previamente el suministro eléctrico. Se muestra en la Figura 10 esta tipología de ascensor (Ingesor, 2018).

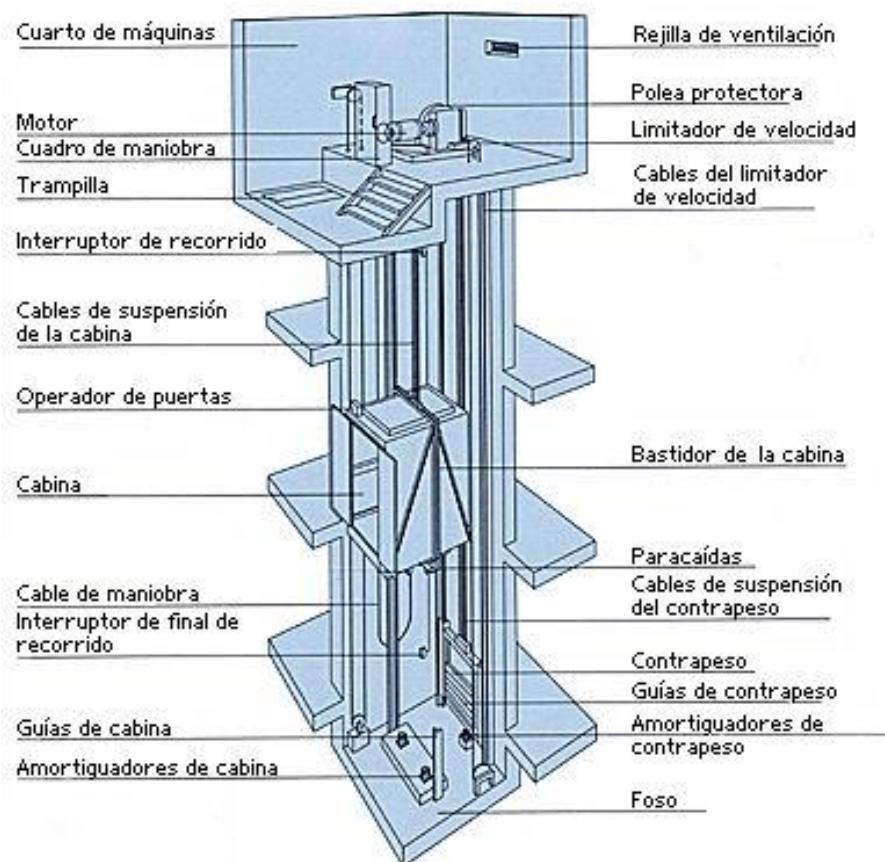


Figura 10: Ascensor electromecánico (Wikimedia Commons)

2.2.2. Según su aspecto de diseño

- **Ascensores panorámicos:**

Son un grupo de elevadores que se conciben como un espacio móvil abierto al exterior, un balcón desde el que contemplar el edificio y su entorno, permitiendo una relación visual del pasajero con el ambiente de forma directa, ya sea en el interior del edificio como en el exterior. Se muestra un ejemplo de ascensor panorámico en la Figura 11 (Grupo de ascensores Enor, 2018).



Figura 11: Ascensor panorámico (Wikimedia Commons)

- **Ascensores multicabina:**

Los sistemas de ascensores multicabina son una apuesta de futuro cuyo objetivo principal es que el usuario que espera la llegada de un ascensor sea atendido en un periodo muy corto de tiempo gracias a que habrá varias cabinas de ascensores en el mismo hueco. El grupo *ThyssenKrupp* es pionero en este tipo de tecnología y ahora trabaja en un grupo de ascensores que denomina MULTI.

El sistema MULTI no emplea ruedas convencionales sobre raíles, sino que hace uso de la tecnología de propulsión lineal magnética. El objetivo del sistema MULTI es crear un bucle continuo en el que las cabinas suben por un conducto y bajan, por el contrario. Se muestra a continuación, en la Figura 12, el sistema de ascensores multicabina que se encuentra en fase de implementación en el edificio *East Side Tower* en la ciudad de Berlín (La nueva era de los ascensores revoluciona la construcción de mediana y gran altura, 2018).



Figura 12: Sistema de ascensores multicabina (TyssenKrupp)

- **Ascensores *Double Deck*:**

Esta tipología de ascensores tiene la peculiaridad de tener dos cabinas conjuntas, una sobre la otra, de modo que ambas recorren el hueco del ascensor solidariamente. Gracias a este novedoso sistema, se consigue desplazar un mayor número de pasajeros por cada viaje realizado consiguiéndose que el tiempo de espera de cada usuario para recibir un ascensor sea menor. No solo se consigue aumentar el confort del pasajero gracias a un menor tiempo de espera, sino que además, al realizarse menor número de viajes para servir la demanda de los pasajeros, el consumo energético es inferior.

Para el empleo de este tipo de ascensores se tienen dos vertientes diferenciadas. En la primera de ellas, se alojará a los pasajeros en la cabina inferior si estos van a plantas impares y se alojarán en la cabina superior aquellos pasajeros cuyo destino son plantas pares. La segunda vertiente consiste en el empleo de una cavidad superior e inferior (no visitables desde el ascensor) en el edificio de modo que el ascensor pueda servir todas las plantas tanto con la cabina superior como inferior. Se puede observar en la Figura 13 y Figura 14 ascensores con este tipo de arquitectura.



Figura 13: Ascensor *Double Deck* (Wikimedia Commons)



Figura 14: Sistema de ascensores *Double Deck* (Wikimedia Commons)

Debido a las ventajosas características de este tipo de ascensores, se pueden encontrar en gran cantidad de edificios. A continuación, se muestran edificios que cuentan con sistemas de ascensores con arquitectura *Double Deck*:

→ **Stratosphere Las Vegas**

Se encuentra este edificio de 350 metros de altura en la ciudad de Las Vegas (Estados Unidos). El objetivo principal de esta construcción reside en su empleo como hotel, casino y restaurante. Además, en la planta última se encuentra la atracción más alta del mundo (*The Big Shoot*). Se muestra en la Figura 15.



Figura 15: Stratosphere, Las Vegas (Wikimedia Commons)

→ **Torre Picasso**

La Torre Picasso se encuentra en la Plaza Pablo Ruiz Picasso 1 (junto al Paseo de la Castellana) en la ciudad de Madrid (España). Consta de 51 plantas repartidas en sus 157 metros de altura para labores de oficina principalmente. Google, Deloitte y Accenture son algunas empresas cuya sede se encuentra en este emblemático edificio madrileño. Cuenta además con sótano y helipuerto. Se muestra en la Figura 16 (Torre Picasso, 2018).



Figura 16: Torre Picasso, Madrid (Wikimedia Commons)

→ **The Shard**

El edificio *The Shard* se encuentra en la zona de Southwark en la ciudad de Londres (Reino Unido). Consta de 72 plantas utilizables (de las 87 plantas que dispone) repartidas en los 309,6 metros de altura que lo hacen el edificio más alto de la Unión Europea y cuyo objetivo principal es el uso como oficinas además de disponer de restaurantes, apartamentos y un hotel. Se observa en la Figura 17 (The Shard, 2018).



Figura 17: The Shard (izquierda) junto a Tower Bridge, Londres (Wikimedia Commons)

→ **Citigroup Center**

Se encuentra el edificio *Citigroup Center* en la ciudad de New York (Estados Unidos), siendo uno de los más altos rascacielos de la ciudad. Cuenta con 279 metros repartidos en sus 59 plantas. Actualmente alberga la sede central del banco *Citibank*. Se muestra en la Figura 18.



Figura 18: Citigroup Center, New York (Wikimedia Commons)

→ Torre Burj Khalifa

Se encuentra en la ciudad de Dubai (Emiratos Árabes Unidos). Sus 828 metros de altura repartidos en 163 plantas hacen que este rascacielos sea el más alto del mundo desde el año 2010 tras arrebatarse dicho puesto a la Torre *Taipei 101* de la República de China. La torre *Burj Khalifa* es usada como hotel, residencias de lujo, gimnasio y suites corporativas. Además, en la planta 124 se encuentra un mirador denominado “*At the top*”. Cuenta con una serie de ascensores de arquitectura *Double Deck*, instalados por *Otis*, que transportan pasajeros a una velocidad de diez metros por segundo logrando ser los ascensores de doble cabina más veloces del mundo. Se muestra en la Figura 19 (Otis en la torre Burj Khalifa, 2018).



Figura 19: Torre Burj Khalifa, Dubai (Wikimedia Commons)

2.3. Funcionamiento del ascensor

En este apartado se describe el modo de funcionamiento de un ascensor. La importancia de este apartado reside en la necesidad de conocer cómo funciona un ascensor para saber en qué momentos se produce el consumo de energía, que posteriormente va a tratar de ser minimizado (CIBSE, 2000).

Los ascensores disponen de un mecanismo de funcionamiento mediante un contrapeso que ayuda al movimiento del ascensor. El contrapeso que se emplea tiene un peso aproximado del 50% de la carga del ascensor cuando este se encuentra a media carga. Es decir, se emplea un sistema de poleas y un motor para el funcionamiento de la cabina. Se muestra en la Figura 20 el esquema de un ascensor, su contrapeso y el motor.

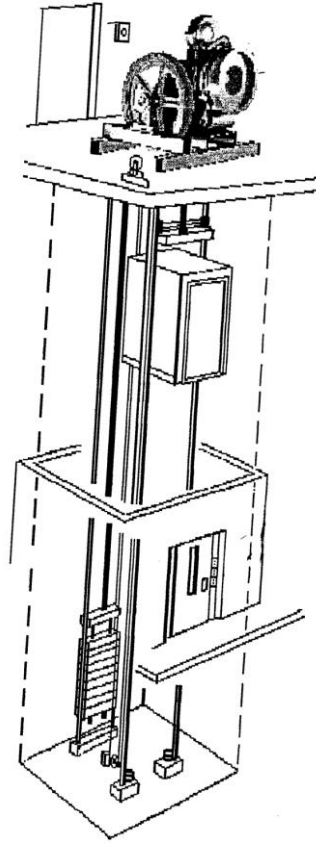


Figura 20: Mecanismo de contrapeso de un ascensor (Wikimedia Commons)

En función de los pasajeros que se monten en la cabina pueden ocurrir las siguientes casuísticas:

- El ascensor tiene que realizar una bajada y el número de pasajeros en el interior de la cabina es superior a la carga del contrapeso. En este caso el ascensor baja mediante la fuerza de la gravedad.
- El ascensor tiene que realizar una bajada y el número de pasajeros en el interior es inferior a la carga del contrapeso. En este caso el ascensor realiza la bajada ayudado por el motor.
- El ascensor tiene que realizar una subida y el número de pasajeros es superior a la carga del contrapeso. El motor debe actuar en este caso para ayudar a que el ascensor se eleve.
- El ascensor tiene que realizar una subida y el número de pasajeros es inferior a la carga del contrapeso. En esta casuística el ascensor realiza la subida gracias a la gravedad que afecta sobre el contrapeso.

A pesar de que el proceso de funcionamiento parece trivial, no lo es debido a que existe un término asociado al movimiento del ascensor que se denomina fricción. La fricción del juego de poleas hace que haya que emplear un nivel de energía superior al que se emplearía en el caso ideal en el que no existiese dicho término. Además, hay que sumar el término de energía necesario para activar la parada de la cabina en el destino y para la apertura y cierre de puertas.

3. TRANSPORTE VERTICAL

Se define el transporte vertical como el movimiento de pasajeros y carga a lo largo de un edificio en su eje perpendicular al suelo. Es decir, el transporte vertical se encarga del movimiento de subida y bajada de un edificio.

En este contexto se encuentra la importancia del estudio de sistemas de transporte vertical en edificios de oficinas por la necesidad de la minimización del tiempo de espera de los trabajadores, así como de cara a la reducción del consumo energético para el beneficio de la empresa. Se tiene para ello que definir en primer lugar los tipos de patrones de tráfico que se tiene en edificios de oficinas y en segundo lugar el modo de proceder para la valoración del servicio prestado por el grupo de cabinas del edificio.

3.1. Patrones de tráfico

El estudio del movimiento de los trabajadores en el edificio es importante debido a que según como sea este habrá que incorporar una cantidad mayor o menor de ascensores, así como llevar a cabo un modo de asignación u otra de cabinas a llamadas.

Se observa en la Figura 21 el patrón de tráfico que sigue un edificio de oficinas según la hora del día. En el eje de abscisas se observa la hora y en el eje de ordenadas la cantidad de tráfico ascendente y descendente en términos de porcentaje total de la población. Se muestran con diferentes colores los diferentes tipos de patrones de tráfico (Cortés *et al.*, 2009).

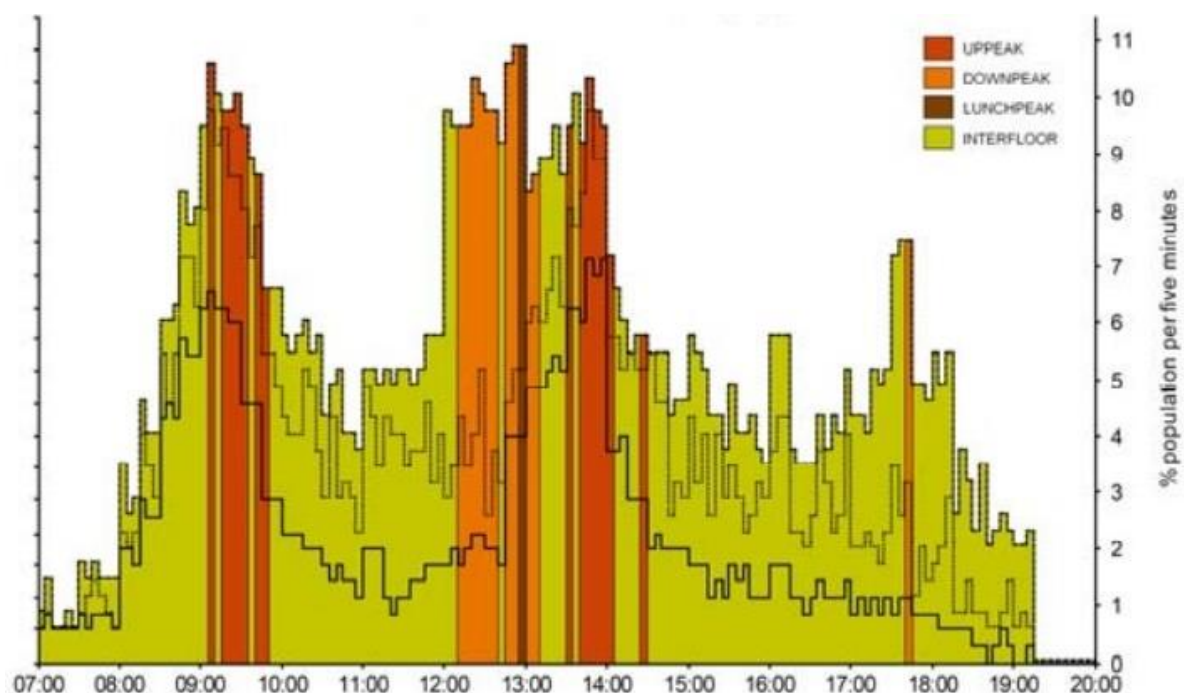


Figura 21: Patrón de tráfico Siikonen (Cortés *et al.*, 2009)

La teoría clásica de Barney (2003) muestra unos patrones de tráfico según la hora del día, siendo estos:

- **Uppeak:**

Se refiere al tramo horario perteneciente al acceso al edificio y por tanto en el cual se producen numerosas llamadas desde la planta baja del edificio con carácter ascendente con el objeto de llegar al puesto de trabajo. Este tipo de tráfico es muy importante debido a la importancia del rápido despacho de pasajeros a plantas del edificio a la hora del comienzo de la jornada laboral.

- **Interfloor:**

Se refiere a los tramos horarios en los cuales existen viajes entre plantas intermedias del edificio. Se divide en dos tráficos diferenciados, aunque ambos se agrupan como *Interfloor*.

- i. **Balanced Interfloor Traffic**, es el tráfico habitual de media mañana o media tarde.
- ii. **Unbalanced Interfloor Traffic**, es el tráfico no habitual de media mañana o media tarde y que suele ser causado debido a alguna reunión de personal.

- **Lunchpeak:**

Se refiere al tramo horario en el cual los empleados del edificio abandonan el mismo para realizar el almuerzo. Está caracterizado por un intenso tráfico de llamadas descendentes a la hora de abandonar el edificio y de llamadas ascendentes al volver del periodo de descanso.

- **Downpeak:**

Se refiere al tramo horario coincidente con la hora de finalización del turno de trabajo. Se caracteriza por recibir un tráfico muy intenso de llamadas descendentes con destino final la planta baja del edificio. Este tipo de tráfico tiene una gran importancia debido a que el requerimiento de cabinas es muy alto ya que gran parte del edificio ha de ser desalojado en un breve lapso.

Se suele considerar el periodo *Uppeak* como el más severo en cuanto al número de llamadas a ascensores debido a que la hora de entrada al puesto de trabajo genera una gran masa de población que accede al edificio. Sin embargo, debido a la flexibilización de los horarios de trabajo y a la prohibición de fumar dentro del mismo, el tráfico *Interfloor* adquiere un gran protagonismo y es estudiado en la actualidad con el objeto de ofrecer una mejor calidad de servicio. De igual modo, el tráfico *Downpeak* y *Lunchpeak* también deben ser estudiados para poder llevar a cabo una optimización completa de recursos (tiempo/coste) a lo largo de todo el periodo de empleo del edificio.

3.2. Nivel de servicio

Para evaluar la calidad del nivel de servicio ofrecido por el despacho de llamadas a cabinas se toman las siguientes Tabla 2 y Tabla 3 como referencia de satisfacción del pasajero en función del tiempo medio de espera de este frente al ascensor.

Tabla 2: Nivel de servicio según porcentaje de llamadas respondidas

Nivel de servicio	Porcentaje de llamadas respondidas en	
	30 segundos	60 segundos
Excelente	>75%	>98%
Bueno	>70%	>95%
Regular	>65%	>92%
Pobre/inaceptable	<65%	<92%

Se observa que, si el 98% de las llamadas a ascensores son servidas en menos de 60 segundos o si el 70% es asignado en menos de 30 segundos, la calidad del servicio es excelente. Se puede marcar, por tanto, en este punto como objetivo un nivel de servicio excelente para la asignación posterior que se realice.

Tabla 3: Nivel de servicio en función del tiempo empleado en responder las llamadas

Nivel de servicio	Porcentaje de llamadas respondidas en	
	50%	90%
Excelente	20 segundos	45 segundos
Bueno	22,5 segundos	50 segundos
Regular	25 segundos	55 segundos
Pobre/inaceptable	>25 segundos	>55 segundos

Se tiene una vertiente de medición de la calidad del servicio mediante la cual se va a tener un nivel de servicio excelente si el 50% de las llamadas son servidas en 20 segundos o si el 90% es servido en 45 segundos.

Mediante las dos tablas anteriores la calidad del servicio se mide a través del tiempo medio de espera que el pasajero se encuentra ante la puerta del ascensor desde que el ascensor ha sido llamado hasta que el ascensor comienza a abrir sus puertas. Este tiempo se denomina mediante las siglas **AWT**, cuyo significado es *Average Waiting Time*. Además de este tiempo, en el presente estudio se va a contabilizar el tiempo en el que el pasajero se encuentra dentro del

ascensor hasta llegar a su destino, denominándose **AJT** (*Average Journey Time*). Por último, de la suma del tiempo medio de espera más el tiempo medio de tránsito resulta el tiempo medio total, denominado **ATT** (*Average Total Time*).

La problemática surge al contabilizar el tiempo medio de espera de los pasajeros sucesivos que llegan ante la puerta del ascensor tras haberse realizado ya la llamada de este. Una de las opciones para solventar dicho problema es contabilizar manualmente el tiempo mediante un cronómetro, pero es una opción poco factible. En todo caso, esto no es inconveniente para la realización del trabajo ya que se emplea el software *Elevate* que tiene en cuenta este tiempo.

4. ASIGNACIÓN DE CABINAS A LLAMADAS

En esta sección se define el problema de asignación de cabinas a llamadas de pasajeros y los diferentes métodos de asignación que se tienen. Además, se caracterizan los métodos de resolución que se han llevado a cabo para la obtención de diferentes soluciones al problema.

4.1. Definición del problema

El problema consiste en un edificio de oficinas con un número de plantas determinado. Se encuentra que el edificio tiene una serie de ascensores con arquitectura *Double Deck* y unos pasajeros que realizan llamadas tanto de subida como de bajada desde las distintas plantas del edificio. Cabe destacar que el número de llamadas a ascensores que realizan los pasajeros dependen del tramo horario como se observa en los distintos patrones de tráfico de la sección anterior.

El objetivo principal que se debe satisfacer es la asignación de las distintas llamadas a ascensores de modo que el tiempo medio de espera de los pasajeros (AWT) sea el menor posible. Además, se tiene un objetivo secundario, relacionado con el principal, que consiste en que el tiempo medio de tránsito (AJT) también sea el menor posible.

Además de lo anterior, de la minimización del tiempo medio de espera y tránsito, así como del menor número de paradas realizadas (Al-Sharif *et al.*, 2017) gracias al empleo de ascensores con arquitectura *Double Deck* se consigue una reducción del consumo energético del edificio. Dicha reducción se debe reflejar en los resultados obtenidos del estudio realizado por lo que forma parte del problema.

Este problema cuenta con una serie de restricciones, tanto generales como particulares que se pasan a detallar a continuación (Cortés *et al.*, 2003).

i. Restricciones generales

Se dividen las restricciones generales en explícitas e implícitas.

- **Restricciones generales explícitas:**

- Solo se puede servir a una llamada por cabina en cada instante temporal.
- La capacidad portante (kg) de la cabina limita el número máximo de pasajeros que pueden montar en la misma al mismo tiempo.
- Solo se realizan paradas por parte de la cabina en caso de que un pasajero requiera subir o bajar del ascensor en esa planta de parada.
- Solo se puede realizar una parada en una planta si no existe otra cabina efectuando el servicio de carga/descarga en dicha planta.

- **Restricciones generales implícitas:**

- Las paradas realizadas por la cabina se corresponden con las llamadas realizadas por los pasajeros de forma secuencial al sentido de la marcha de la cabina, sin omitir ninguna planta.
- La direccionalidad del ascensor en servicio no puede cambiar hasta servir a todos los pasajeros que hayan solicitado dicha dirección.

- El algoritmo va a tomar la decisión de subir en caso de que haya ambigüedad entre subir/bajar.

ii. Restricciones particulares

Las restricciones particulares dependen de las condiciones específicas del edificio. Se detallan algunas restricciones particulares que se pueden encontrar:

- Se debe disponer de al menos un ascensor para minusválidos.
- Empleo de algún ascensor de forma privada para algún determinado colectivo.
- Disponer de uno o varios ascensores para viajes especiales, tales como la subida al mirador en rascacielos.
- Disponer de plantas con especial preferencia que deben ser servidas con una mayor brevedad que el resto.

4.2. Métodos de asignación

Una vez se ha definido el problema con sus restricciones y la función objetivo que se quiere lograr se ha de caracterizar el modo de evaluar cada solución obtenida. Esto es, cada solución que contiene la asignación de llamadas a cabinas debe ser evaluada para poder obtener la función objetivo. Se han llevado a cabo dos métodos de asignación de cabinas a ascensores diferentes que se detallan a continuación.

4.2.1. Asignación basada en distancia

La asignación de cabinas a ascensores mediante criterios de distancia se basa en el cálculo de la función objetivo como la suma de las distancias desde las cabinas que se emplean hasta las plantas desde las cuales se realizan llamadas. Gracias a la minimización de la distancia total, se consigue la reducción del tiempo medio de espera y tránsito de los pasajeros (Vázquez Ledesma, 2016).

El modo de proceder para el cálculo sigue el diagrama de flujo que se muestra en la Figura 22.

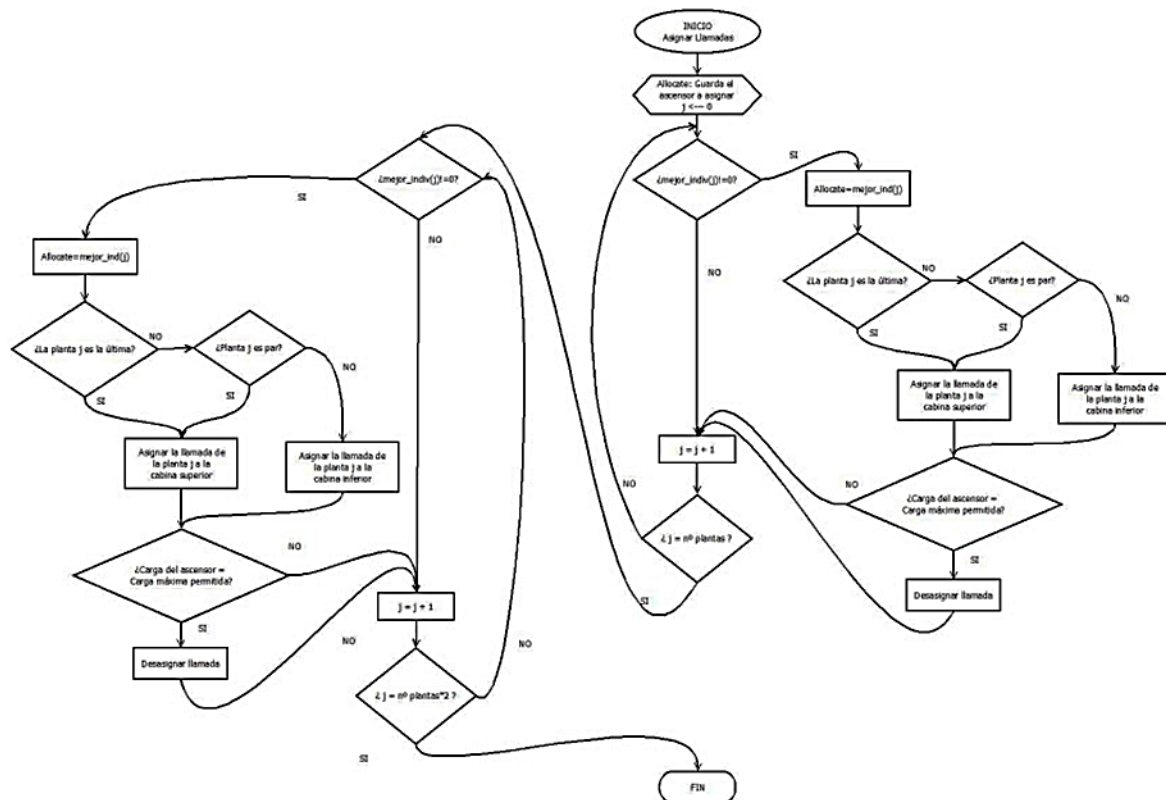


Figura 22: Asignación de llamadas a cabinas basada en distancia

Además, hay que tener en cuenta una serie de factores importantes para el cálculo de la distancia:

- En el caso de que el ascensor se encuentre bajando. Si se recibe una llamada de subida desde una planta, la distancia se calcula como la suma de la distancia desde la posición actual hasta la planta más baja por servir más la distancia desde dicha planta más baja hasta la planta de llamada. Esto es debido a la restricción de no cambio de direccionalidad.
- En el caso de que el ascensor se encuentre subiendo. Si se recibe una llamada de bajada desde una determinada planta, la distancia se calcula como la suma de la distancia desde la posición actual hasta la planta más alta por servir más la distancia desde dicha planta más alta hasta la planta de llamada. Se debe a la restricción de no cambio de direccionalidad.
- En el caso de que el ascensor se encuentre bajando. Si se recibe una llamada de bajada, se comprueba si es posible parar el ascensor en dicha planta. En caso de que no sea posible la parada, la distancia se calcula como la suma de la distancia desde la planta actual hasta la planta más baja por servir más la suma de la distancia desde dicha planta baja hasta la planta más alta por servir más la distancia desde la planta alta hasta la planta de llamada.
- En el caso de que el ascensor se encuentre subiendo. Si se recibe una llamada de subida, se comprueba si es posible parar el ascensor en dicha planta. En caso de que no sea posible la parada, la distancia se calcula como la suma de la distancia desde la planta actual hasta la planta más alta por servir más la suma de la distancia desde dicha planta alta hasta la planta más baja por servir más la distancia desde la planta baja hasta la planta de llamada.

4.2.2. Asignación basada en tiempo

La asignación de cabinas a llamadas mediante criterios de tiempo se basa en el tiempo que el pasajero se encuentra frente al ascensor (AWT) y el tiempo que el pasajero se encuentra viajando hasta llegar a su destino (AJT). Se calcula por tanto la suma del tiempo total que los pasajeros pasan frente al ascensor y viajando hacia las plantas destino. Se muestra en la Figura 23 el modo de proceder para el cálculo de la función objetivo mediante este tipo de asignación (Reyes López, 2017).

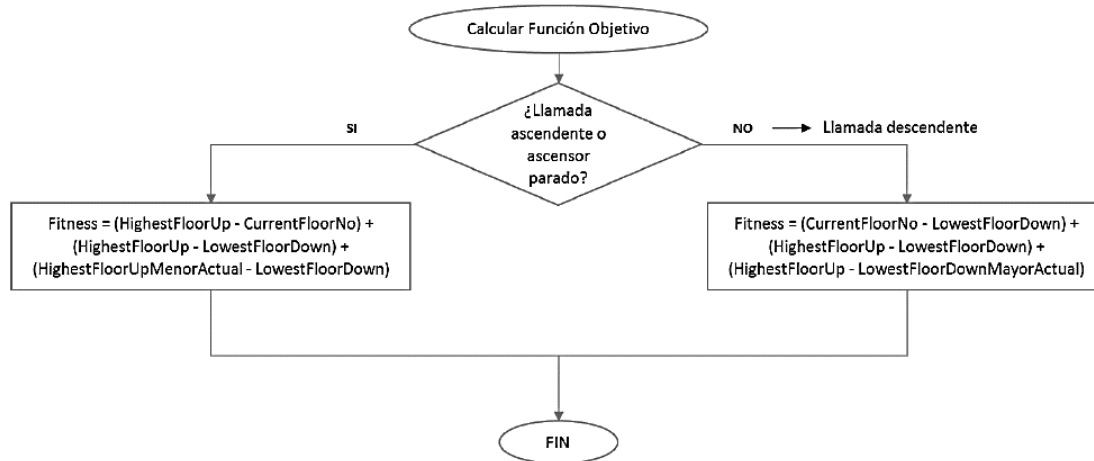


Figura 23: Asignación de cabinas a llamadas basada en tiempo

De igual modo que ocurre con la asignación basada en distancia, existen una serie de características especiales de diseño para el cálculo del tiempo total:

- En el caso de que el ascensor se encuentre bajando. Si se recibe una llamada de subida desde una planta, la distancia se calcula como la suma de la distancia desde la posición actual hasta la planta más baja por servir más la distancia desde dicha planta más baja hasta la planta de llamada. Una vez se ha obtenido la distancia, esta se multiplica por la velocidad media del ascensor para obtener el tiempo.
- En el caso de que el ascensor se encuentre subiendo. Si se recibe una llamada de bajada desde una determinada planta, la distancia se calcula como la suma de la distancia desde la posición actual hasta la planta más alta por servir más la distancia desde dicha planta más alta hasta la planta de llamada. Una vez se ha obtenido la distancia, esta se multiplica por la velocidad media del ascensor para obtener el tiempo.
- En el caso de que el ascensor se encuentre bajando. Si se recibe una llamada de bajada, se comprueba si es posible parar el ascensor en dicha planta. En caso de que no sea posible la parada, la distancia se calcula como la suma de la distancia desde la planta actual hasta la planta más baja por servir más la suma de la distancia desde dicha planta baja hasta la planta más alta por servir más la distancia desde la planta alta hasta la planta de llamada. Una vez se ha obtenido la distancia, esta se multiplica por la velocidad media del ascensor para obtener el tiempo.
- En el caso de que el ascensor se encuentre subiendo. Si se recibe una llamada de subida, se comprueba si es posible parar el ascensor en dicha planta. En caso de que no sea posible la parada, la distancia se calcula como la suma de la distancia desde la planta actual hasta la planta más alta por servir más la suma de la distancia desde dicha planta alta hasta la planta más baja por servir más la distancia desde la planta baja hasta la planta de llamada. Una

vez se ha obtenido la distancia, esta se multiplica por la velocidad media del ascensor para obtener el tiempo.

Es posible una variante que consiste en no contabilizar la distancia y multiplicar por la velocidad, sino contabilizar el número de plantas que se han de recorrer y multiplicar por el tiempo medio estimado para recorrer la distancia que hay de una planta a otra.

Además del tiempo anterior que solo tiene en cuenta la posición relativa del ascensor y de las plantas a servir, se tienen en cuenta una serie de tiempos que solo dependen del ascensor y los pasajeros. Dichos tiempos se muestran en la Tabla 4 y se aplican sumándolos a cada parada que el ascensor realiza. Se consigue mediante este último sumando que, al minimizar el tiempo, el número de paradas sea menor ya que con eso se consigue no incurrir en los tiempos de la tabla.

Tabla 4: Tiempos asociados al cálculo de la asignación basada en tiempo

Concepto	Duración
Tiempo de apertura de puertas	2 segundos
Tiempo de cierre de puertas	2 segundos
Tiempo de aceleración	2 segundos
Tiempo de desaceleración	2 segundos
Tiempo de bajada/subida de pasajeros	5 segundos

4.3. Métodos de resolución

La optimización se ha convertido en los últimos años en una prioridad para las empresas debido a que cada vez el coste de los recursos es mayor, pudiendo ser escasos incluso, por lo que es necesario conseguir una eficiencia capaz de lograr los mejores resultados con una cantidad de recursos mínima.

El empleo de técnicas de optimización es necesario y va evolucionando en el tiempo. Se trata de conseguir la mejor solución en un tiempo considerado como factible, esto es, si se tiene un problema operacional se requiere de una solución rápida, no siendo válido que se tarde meses en obtener una buena solución.

Ya que encontrar la solución óptima de un problema no es trivial, se emplean métodos de resolución que aproximan al valor óptimo (o incluso lo pueden llegar a obtener). A este tipo de técnicas se les conoce como metaheurísticas.

Se encuentran metaheurísticas que se basan en la observación de la naturaleza para su semejanza posterior con el problema a optimizar, entre ellas se encuentran el “Algoritmo genético” o el “Algoritmo de colonias de hormigas”. Otras metaheurísticas se basan en vecindades para explorar las distintas posibles soluciones, entre las que se encuentran el “Algoritmo de recocido simulado” o el “Algoritmo de búsqueda Tabú”.

Se muestra en los siguientes apartados los diferentes métodos de resolución que se han implementado para la asignación de cabinas a llamadas y su posterior comparación de resultados en función del método de resolución empleado.

4.3.1. Búsqueda Tabú

Proveniente del inglés (*tabú search*), la búsqueda tabú es una metaheurística que guía un algoritmo heurístico de búsqueda local para explorar el espacio de soluciones más allá de la simple optimalidad local, según Fred Glover, su primer definidor (Glover, 1989).

La búsqueda tabú se basa en tres pilares:

- Empleo de memoria adaptativa para permitir evaluar la información de búsqueda histórica.
- Empleo un mecanismo de memoria que restringe y libera el proceso de búsqueda.
- Empleo de memorias en distintos lapsos de tiempo (corto, medio y largo) de modo que se puedan guardar aquellas características que logran una buena solución.

Es importante la distinción entre memoria a corto y largo plazo. La memoria a corto plazo almacena los atributos de solución que han sido cambiados en el pasado reciente por lo que también recibe el nombre de memoria basada en hechos recientes. Estos atributos seleccionados recientemente se designan como “tabú-activos” y las soluciones que los contienen (o combinaciones particulares de estos atributos) pasan a formar parte de la lista tabú para evitar que pertenezcan a la futura vecindad y puedan llegar a ser revisitadas. El tiempo que un atributo permanece como “tabú-activo” se denomina tenencia tabú y se mide en número de iteraciones, es variable según los atributos y las combinaciones de estos. En cambio, la memoria a largo plazo emplea memoria basada en frecuencia de cambio de atributos con la cual se pretende potenciar la búsqueda de soluciones, aunque a veces no es necesario su empleo.

No siempre el que una solución se encuentre en la lista tabú indica que dicha solución no sea válida, ya que el método define unos niveles de aspiración con los cuales se evalúan dichas soluciones en la función objetivo y si los resultados obtenidos son considerados como admisibles, por dar mejor solución que la anterior generada, estos son rescatados de la lista tabú.

El algoritmo de búsqueda tabú emplea estrategias de intensificación y/o diversificación. Siendo la estrategia de intensificación aquella en la que la evolución de soluciones es favorecida por combinaciones de movimiento y características de soluciones que históricamente hayan sido buenas. Sin embargo, la estrategia de diversificación buscará explorar nuevas regiones a través del cambio de las reglas de elección de atributos. Se busca el empleo de atributos que no hayan sido usados frecuentemente.

Para la resolución del problema de asignación de llamadas a cabinas mediante la metaheurística de búsqueda tabú se emplea el diagrama de flujo de la Figura 24.

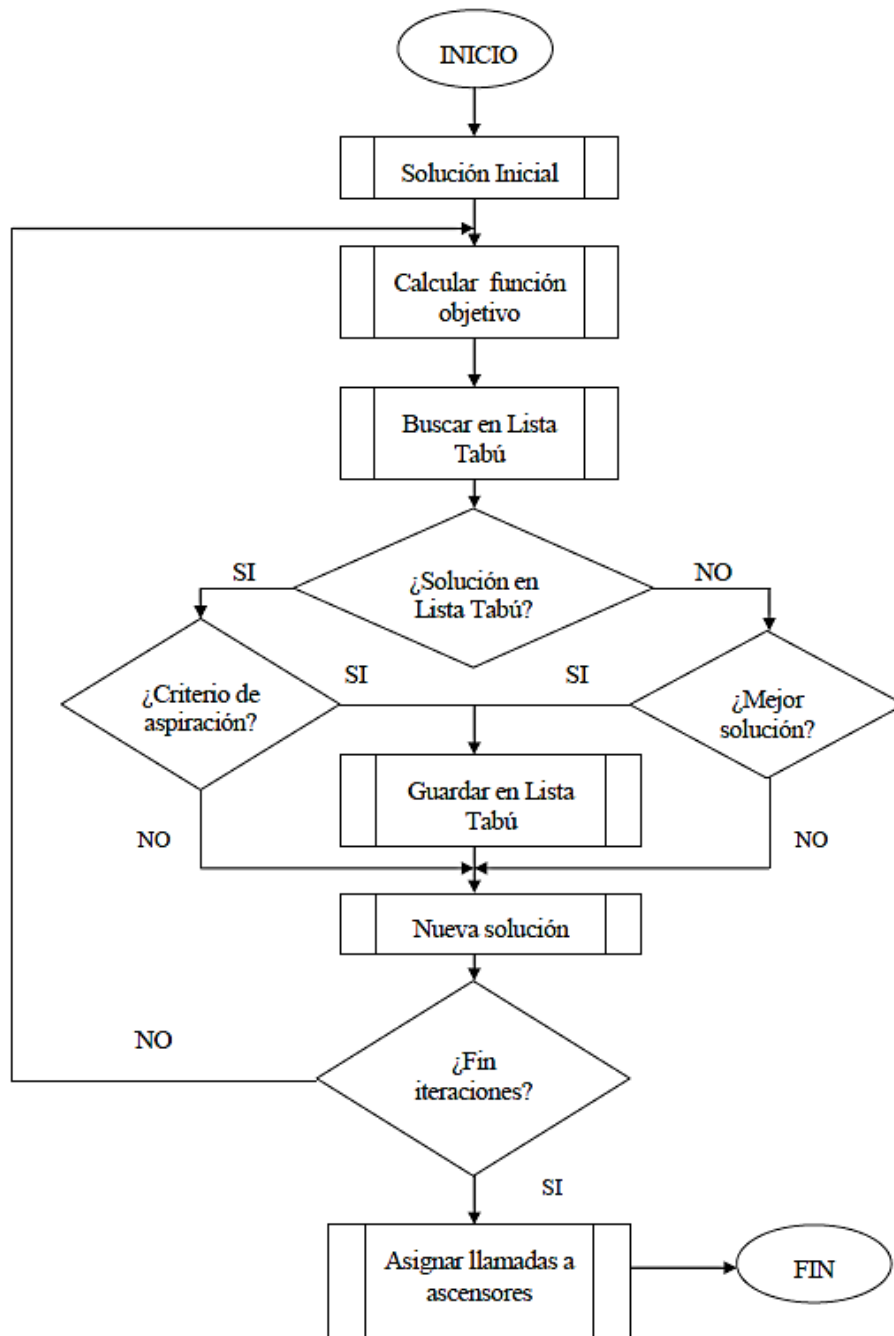


Figura 24: Diagrama de flujo Búsqueda Tabú

El algoritmo que se implementa en C++ recibe del software *Elevate* las características del edificio y de los ascensores, así como del patrón de tráfico del edificio que se va a emplear. Una vez se procesa la información mediante el algoritmo programado, se devuelven a *Elevate* las cabinas que deben ir a cada una de las plantas que reciben llamadas de pasajeros. Se detalla a continuación cada uno de los pasos que se llevan a cabo en el algoritmo de Búsqueda Tabú programado.

4.3.1.1. Solución inicial

Para iniciar el algoritmo, se busca una solución inicial de asignación de llamadas a cabinas en la que se rellena el vector que contiene los ascensores (1 si el ascensor “i” va a ser empleado y 0 si el ascensor “i” no será empleado) de forma aleatoria. Además, las cabinas que se van a emplear son asignadas a plantas con llamadas de manera aleatoria.

4.3.1.2. Calcular la función objetivo

Se trata del cálculo de la suma de distancias o tiempos de cada ascensor, contabilizando solo los ascensores que van a ser empleados. El cálculo a realizar se muestra en el apartado 4.2. y va a depender de si la asignación se realiza en base al tiempo o en base a la distancia.

4.3.1.3. Buscar en la Lista Tabú

Una vez se tiene una solución propuesta y conocida su función objetivo asociada, se busca en la Lista Tabú dicha solución para comprobar si ya se ha visitado. En caso de que la solución propuesta se encuentre en la Lista Tabú, se comprueba si se cumple el criterio de aspiración. Es decir, se acepta aquella solución cuya función objetivo actual tome un valor un 20 % por debajo de la mejor función objetivo encontrada hasta el momento. En caso de que la solución propuesta no se encuentre en la Lista Tabú, si su función objetivo es mejor que la mejor función objetivo encontrada hasta ese punto, se acepta como posible solución al problema.

4.3.1.4. Guardar en la Lista Tabú

La Lista Tabú se encuentra formada por una matriz en la cual se guarda por cada fila el vector de ascensores que se emplean y su función objetivo correspondiente. La capacidad de dicha lista viene dada por un término denominado *Tabú Ténere* y es fijado al comienzo del uso del algoritmo. Cada vez que se va a incorporar una nueva solución a la Lista Tabú, se elimina la solución más antigua de la lista de forma que se sigue un proceso *First In First Out* (FIFO).

4.3.1.5. Nueva solución

La generación de una nueva solución viene definida por tres posibles casuísticas:

1. Se genera una solución que se encuentra en la Lista Tabú y que no cumple el criterio de aspiración.
2. Se genera una solución que no se encuentra en la Lista Tabú pero no tiene una función objetivo mejor que la actual.
3. Se guarda la solución actual en la Lista Tabú gracias a haber cumplido el criterio de aspiración o mejorar la mejor función objetivo y se requiere de una nueva solución para seguir la búsqueda de una mejor solución.

El proceso llevado a cabo para la creación de una nueva solución consiste en la asignación al vector de ascensores que se emplean el valor contrario al que se tenía anteriormente si se cumple una verdad aleatoria. Es decir, si anteriormente el ascensor “*i*” se empleaba (valor 1), si se cumple que un número aleatorio (entre 0 y 1) es mayor que 0.5, el ascensor “*i*” pasa a no emplearse (valor 0). De igual modo ocurre con aquellos ascensores cuyo valor actual es 0. Se consigue mediante este modo de generación de vecindario la reducción del mismo de modo que se agiliza en gran medida la resolución del problema (siendo esto muy necesario pues no se puede emplear un gran tiempo computacional en la asignación de llamadas a ascensores).

4.3.1.6. Asignación de llamadas a ascensores

Una vez han finalizado las iteraciones del proceso de búsqueda de soluciones, se escoge aquella solución cuya función objetivo es menor. Es decir, aquella que consigue que la suma de las distancias o tiempos desde los ascensores que se van a emplear hasta las plantas que tienen llamadas sea la menor posible.

Los ascensores que se emplean, dados por la mejor solución obtenida, se encuentran asociados a cada una de las llamadas del edificio a las que van a servir mediante una variable llamada $P(j)$ que toma el valor i en caso de que la planta j del edificio sea servida por la cabina i . Se almacena la solución de ascensores que se emplean en las variables *Allocate* y *Allocated* mediante un bucle de recorrido de dos veces el número de plantas del edificio ($NoFloors \cdot 2$) en el que, si la planta j ha recibido una llamada, se guarda en la variable *Allocate* o *Allocated* (según sea la llamada ascendente o descendente) el valor de la variable $P(j)$. Es decir, se guarda el ascensor i que ha quedado asignado a esa planta. En la primera mitad del bucle se almacenan las llamadas ascendentes en la variable *Allocate* y en la segunda mitad las llamadas descendentes en la variable *Allocated*.

Debido a que los ascensores empleados tienen arquitectura *Double Deck* la asignación de cabinas no es trivial ya que una vez se ha asignado un ascensor a una planta hay que asignar con las variables de *Elevate* la cabina superior ($l[Allocate/d].m_UpLandingCallsUpperCar$ o $l[Allocate/d].m_DownLandingCallsUpperCar$ igual a 1) a las llamadas de plantas pares y la última del edificio y la cabina inferior ($m_UpLandingCalls$ o $m_DownLandingCalls$ igual a 1) a las llamadas de plantas impares y la primera del edificio.

Se realiza un bucle con el número de plantas ($NoFloors$) multiplicado por 2 ya que en la primera mitad se registran las llamadas ascendentes y en la segunda mitad las llamadas descendentes. A cada planta con llamada, esto es, cuando las variables $m_UpLandingCalls$ o $m_DownLandingCalls$ toman el valor 1, se le ha asignado un ascensor que viene dado por las variables *Allocate* (llamadas ascendentes) y *Allocated* (llamadas descendentes) anteriormente citadas.

Una vez se tiene la planta con llamada y el ascensor que se le ha asignado se consulta si la planta a servir es la última del edificio. En caso afirmativo, se le asigna la cabina superior. En el caso contrario, se consulta si la llamada es par en cuyo caso se le asigna la cabina superior. Si la planta es impar o la llamada es recibida de la planta baja del edificio se asigna la cabina inferior.

Además, si tras realizar la asignación de una cabina a una planta, se detecta que la cabina sobrepasaría el límite de su capacidad portante, la cabina debe ser desasignada para atender a esa planta a la que se le asignó.

Se muestra en la Figura 25 el diagrama de flujo empleado para la asignación de cabinas con arquitectura *Double Deck* a llamadas de pasajeros desde las plantas del edificio.

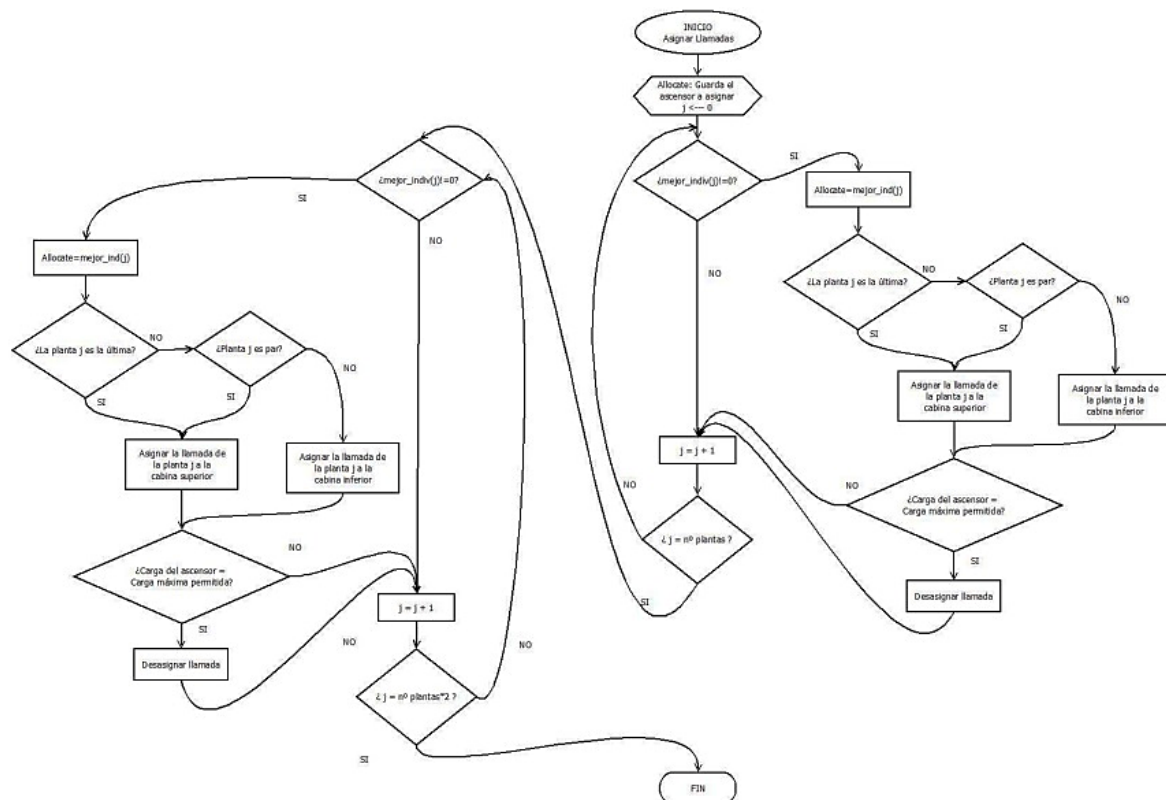


Figura 25: Asignación de llamadas a ascensores

4.3.2. Algoritmo Genético

Los algoritmos genéticos emplean una tipología de búsqueda que se basa en la teoría de la evolución de Darwin. Concretamente, se basan en los mecanismos de selección natural y genética. El algoritmo genético trata de que cada generación se cree con un conjunto de nuevos individuos utilizando solo aquellas partes adecuadas del progenitor.

Se parte de una población compuesta por unos individuos iniciales. Entre los individuos de la población se producen cruces que dan lugar a otros individuos. Cada individuo tiene asociado un valor relacionado con la bondad de una solución factible del problema. Es decir, el grado de efectividad de dicho individuo para sobrevivir. Una vez se ha producido el cruce, uno de los tres individuos (“padre1”, “padre2” o “hijo”) debe salir de la población. Las posibilidades de salir de la población las tendrá el más débil, o lo que es lo mismo, el que presente una bondad de solución peor (Mesa Jiménez, 2015).

4.3.2.1. Solución inicial

Se crea una población inicial compuesta por un número determinado de individuos. Cada individuo generado se encuentra compuesto por una cantidad de genes igual al doble del número de plantas del edificio para que en la primera mitad se almacenen las llamadas ascendentes y en la segunda mitad las llamadas descendentes.

4.3.2.2. Mutación y cruce

Una vez se ha generado la población inicial, el siguiente paso consiste en la elección de los progenitores. Se escogen dos padres mediante la realización de un torneo en el que se compara la bondad de los posibles padres para escoger los mejores.

Tras obtener los progenitores, se lleva a cabo el cruce de genes de ambos con una probabilidad del 85%. Dicho cruce consiste en la generación del hijo mediante los genes de los padres, teniéndose una probabilidad del 50% de que el hijo herede el gen del padre. Se muestra en la Figura 26 el diagrama de flujo del cruce de individuos.

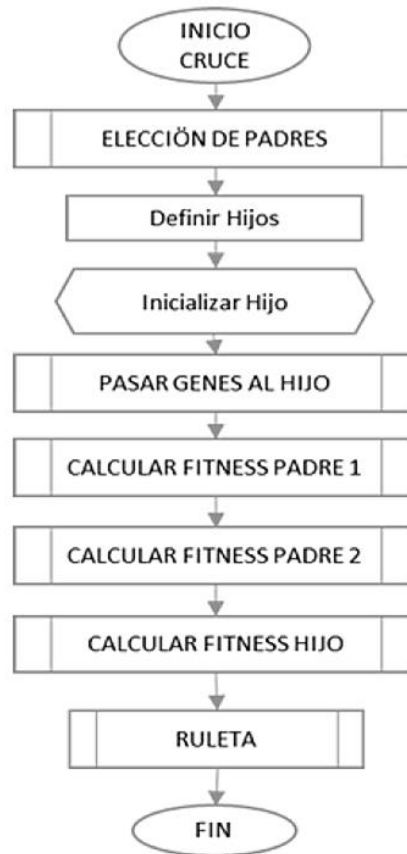


Figura 26: Cruce de individuos

En cambio, con una probabilidad del 15% se tiene que el hijo se genera mediante una mutación. Mediante la mutación, el hijo generado es fruto del intercambio de la posición de los genes de uno de los padres de modo que se obtiene un hijo de características especiales al que se podría llegar realizando un cruce. Se muestra en la Figura 27 el diagrama de flujo de la mutación.

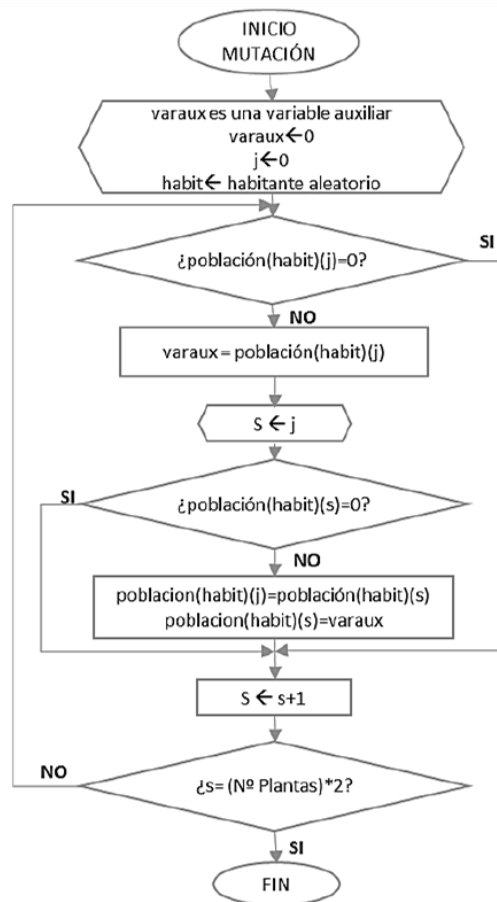


Figura 27: Mutación de individuos

Finalmente, se asigna una probabilidad de salir de la población a cada individuo (ambos padres e hijo) de forma proporcional al *fitness* que cada uno de ellos tiene. Se eliminará de la población aquel individuo con mayor probabilidad de abandonarla.

4.3.2.3. Cálculo del *fitness*

El cálculo de la función objetivo, *fitness*, depende de si se calcula en base a la distancia o al tiempo. Concretamente, se especifica en el apartado 4.2. el modo de proceder para el cálculo de dicha función objetivo. Para el caso del algoritmo genético se realiza el cálculo del *fitness* para toda la población de cara a poder escoger mediante el torneo anteriormente citado a los individuos que obtuvieron el mejor valor.

4.3.2.4. Asignación de llamadas a ascensores

Una vez se tiene el individuo con el mejor *fitness* que se ha encontrado tras el proceso iterativo de búsqueda de soluciones se asignan las cabinas de ascensores a las llamadas siguiendo el diagrama de flujo de la Figura 25 del apartado 4.3.1.6. Se tiene por tanto que la asignación de llamadas a ascensores no depende del método de resolución ya que, tanto para el algoritmo genético como para el algoritmo de búsqueda tabú, la asignación se realiza del mismo modo.

4.3.3. Destination Control Elevate

El software *Elevate 8* implementa entre sus algoritmos resolutivos uno respectivo al funcionamiento de los edificios que emplean ascensores con arquitectura *Double Deck*. Mediante el algoritmo *DC-Elevate (Destination Control- Elevate)* se asignan las llamadas de personas esperando frente al ascensor a cabinas en función de la cercanía de la propia cabina a la planta por servir. Es decir, se sirve de modo preferente a aquellas personas que se encuentran en el ascensor de forma que si se tiene que recoger a un individuo en la planta 20 del edificio pero los pasajeros interiores se bajan en las plantas 17 y 18, el ascensor va a realizar las paradas en las plantas 17 y 18 antes de recoger pasajeros en la planta 20.

5. EXPERIMENTACIÓN

Una vez se ha definido la tipología de ascensores objeto del estudio, las características principales del tráfico en el transporte vertical, así como los métodos de asignación de cabinas a llamadas y los métodos de resolución, se muestran en este apartado las características de la experimentación que se va a llevar a cabo. Además, se pueden observar los resultados obtenidos para cada tipo de tráfico (Interfloor, Lunchpeak y Downpeak).

Es importante reseñar que no se ha realizado el estudio para el tipo de tráfico Uppeak debido a que los algoritmos programados se encuentran diseñados para los otros tres tipos de tráfico. Esto es resultado de una programación mediante la cual los ascensores no son enviados al finalizar su servicio a la planta baja del edificio, siendo esta la problemática por la cual los algoritmos programados no funcionan con solvencia para este tipo de tráfico. No obstante, la cantidad de estudios del tráfico Uppeak para ascensores con arquitectura *Double Deck* es amplio debido a la bondad en las soluciones obtenidas para esta tipología de ascensores y tráfico. En cambio, para los otros tres tipos de tráfico no se tienen muchos estudios para este tipo de cabinas *Double Deck*.

5.1. Características de la experimentación

Se definen en este apartado las características de la experimentación llevada a cabo mediante el software *Elevate*. Para ello, se muestran las características de los edificios, ascensores y tipo de tráfico empleados en las simulaciones objeto de estudio del proyecto.

5.1.1. Edificios

Se emplean edificios con diferente número de plantas de modo que permitan observar el comportamiento de cada algoritmo cuando la altura del edificio se incrementa. Las características de los edificios se muestran en la Tabla 5, en la cual la variable “x” toma los valores: 12, 20, 24, 28, 32, 36 y 40 en función del número de plantas del edificio.

Tabla 5: Características del edificio

Características	Edificio
Número de plantas	x
Separación entre plantas	3,8 metros
Altura del edificio	3,8·x
Personas por planta	50
Población del edificio	50·x

Ya que se emplean ascensores con arquitectura *Double Deck*, los edificios que se emplean en la experimentación cuentan con un número par de plantas debido a que este tipo de ascensores requiere ir siempre de plantas pares a pares y de impares a impares, a no ser que se establezcan dos plantas auxiliares (sin población alguna) en la parte superior e inferior del edificio. Mediante

esta adición de dos plantas auxiliares se permitiría que el ascensor pueda acceder a la primera planta con la cabina superior y a la última planta con la cabina inferior.

5.1.2. Patrones de tráfico

Se van a emplear para la experimentación tres tipos diferentes de tráfico.

Para Interfloor se va a emplear un tipo de tráfico constante (*Constant Traffic (% Building pop per 5 mins)*). Debido a que el patrón de tráfico *Interfloor* ha adquirido un gran protagonismo en la actualidad debido a la flexibilización de los horarios y la prohibición de fumar dentro del edificio se cuenta con que en el edificio existe un 30% de pasajeros que acceden, un 30% de pasajeros que salen del edificio y un 40% de pasajeros que realizan movimientos entre plantas. Se va a suponer también que el uso de las escaleras en el edificio es nulo.

Para Downpeak se va a emplear un tipo de tráfico constante (*Constant Traffic (% Building pop per 5 mins)*). Se cuenta con un 90% de pasajeros que desalojan el edificio con planta destino la planta baja. El 10% de pasajeros restantes se dividen en parte iguales para realizar viajes entre plantas y de entrada al edificio. De igual modo que antes, el uso de escaleras se considera nulo.

Para Lunchpeak se emplea un tipo de tráfico implementado en *Elevate* que se basa en la guía CIBSE. Dicho patrón de tráfico se denomina “*Peters Research (CIBSE) modern office lunch peak 2015*” y se basa en el tipo de tráfico lunchpeak que se experimenta en los edificios modernos de oficinas.

Además, hay una semilla aleatoria (porcentaje del número de pasajeros en cada planta) que varía según el número de ascensores del edificio con el objetivo de mantener un flujo de pasajeros regular. Dicha semilla toma los valores:

Demand (% pop per 5 mins): 12,5 15 17,5 20 22,5

5.1.3. Ascensores

El número de ascensores que se emplean varía según el número de plantas del edificio. Las características de los ascensores se muestran en la Tabla 6.

Tabla 6: Características de los ascensores

Capacidad de los ascensores	1000 kilogramos
Velocidad de los ascensores	2,5 metros/segundos
Aceleración de los ascensores	0,8 metros/segundos ²
Retraso de inicio de marcha	0,5 segundos
Tiempo de apertura de puertas	1,8 segundos
Tiempo de cierre de puertas	2,9 segundos
Tiempo de carga y descarga	1,2 segundos
Factor de capacidad	80%

5.2. Resultados de la experimentación

Se muestran en este apartado los resultados obtenidos para cada tipo de tráfico (Interfloor, Lunchpeak y Downpeak). Concretamente se expone el tiempo medio de espera (AWT), el tiempo medio de tránsito (AJT) y el tiempo medio total de viaje (ATT). Además, también se muestra el consumo energético del edificio obtenido para simulaciones realizadas con una combinación media de número de plantas/número de ascensores.

Los resultados que se muestran para los tiempos y consumos energéticos se extraen de las gráficas que se generan tras realizar las simulaciones mediante el software *Elevate*. Se muestra en la Figura 28 y Figura 29 las gráficas del tiempo medio de espera y tiempo medio de tránsito para la simulación realizada mediante la función de evaluación basada en distancia para un tipo de tráfico Downpeak y un algoritmo genético de resolución.

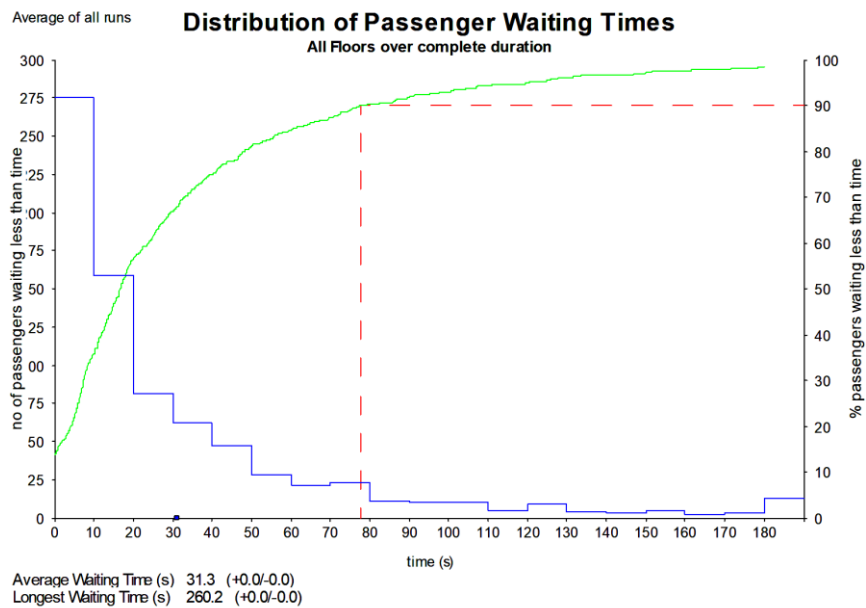


Figura 28: Ejemplo gráfica AWT

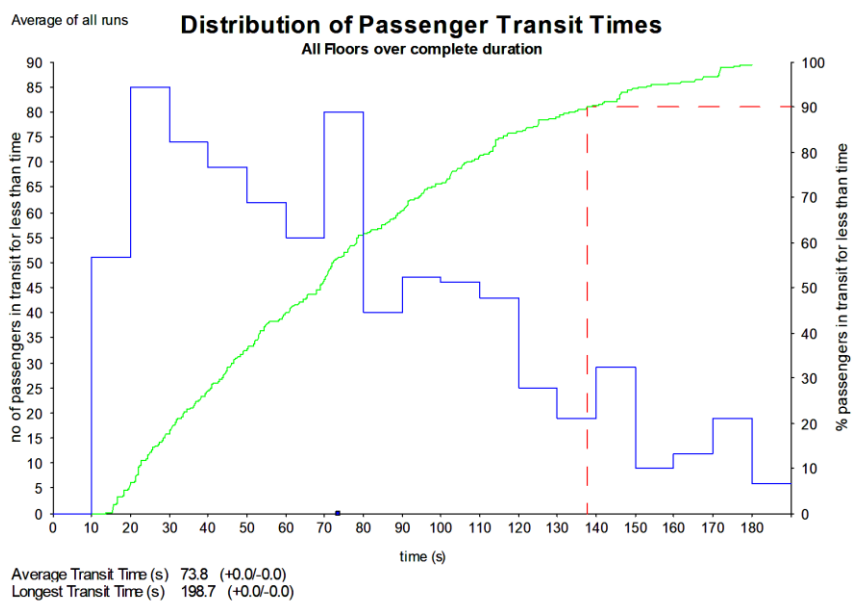


Figura 29: Ejemplo gráfica AJT

Se observa en el lado izquierdo de las gráficas, en el eje de abscisas, el número de pasajeros que esperan menos tiempo que el tiempo que se encuentra en el eje de ordenadas. En el lado derecho, se encuentra el porcentaje de pasajeros que esperan menos que el tiempo dado en el eje de ordenadas.

La línea verde de la gráfica indica la evolución del tiempo de espera de los pasajeros, mientras que las líneas azul y roja indican respectivamente la cantidad y porcentaje de personas que esperan menos tiempo que el marcado en el eje del tiempo.

En este caso concreto se observa que el 90% de los pasajeros esperan menos de aproximadamente 80 segundos. El tiempo medio de espera es de 31,3 segundos. El mayor tiempo de espera que se ha dado en esta simulación es de 260,2 segundos. Con respecto al tiempo de tránsito se tiene que el 90% de los pasajeros han realizado el viaje en menos de 140 segundos mientras que el tiempo medio ha sido de 73,8 segundos.

Para el consumo energético, se emplean gráficas como la que se muestra en la Figura 30 en la que se muestra el consumo energético para la función basada en distancia de un tipo de tráfico Downpeak con algoritmo de búsqueda tabú como método de resolución. Se observa en rojo la evolución del consumo energético, así como el consumo total acumulado en el periodo temporal.

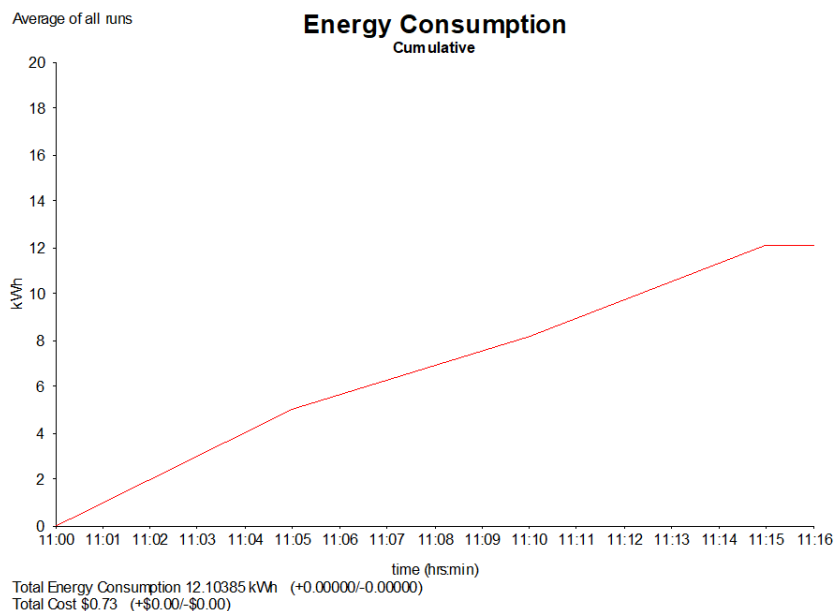


Figura 30: Ejemplo consumo energético

5.2.1. Tráfico Interfloor

Se presentan en este apartado los resultados obtenidos para el estudio del tráfico interfloor empleando los diferentes métodos de resolución. Se muestra en la Tabla 7 los tiempos obtenidos para la función de evaluación basada en distancia y en la Tabla 8 los tiempos obtenidos para la función de evaluación basada en tiempo.

Tabla 7: Tiempos medios de espera, tránsito y total para función de evaluación de distancia. Interfloor

Interfloor		GA			TS			DC-Elevate		
Nº Plantas	Nº Ascensores	AWT	AJT	ATT	AWT	AJT	ATT	AWT	AJT	ATT
12	2	44,3	46,6	90,9	53	47,6	100,6	48,5	48,7	97,2
	5	16,5	38,5	55	15,8	29,4	45,2	22,7	28	50,7
	7	11,3	33,9	45,2	11,5	28,9	40,4	13,6	21,9	35,5
20	4	51,1	67,1	118,2	57,6	67,9	125,5	52	52	104
	8	28,8	59,3	88,1	21,8	45,5	67,3	35,9	37,6	73,5
	11	15,4	45,2	60,6	17,2	39,4	56,6	29,1	30,2	59,3
24	5	51,3	82,3	133,6	60,7	77,6	138,3	55,5	55,9	111,4
	10	23,5	70,3	93,8	25,8	54,1	79,9	38,8	42,3	81,1
	13	17,1	51,7	68,8	18,8	50,1	68,9	26,8	33,1	59,9
28	6	57	97,2	154,2	62,6	91,8	154,4	62,7	62,3	125
	13	19,7	77,3	97	24	58,1	82,1	45,2	44,2	89,4
	16	18,2	58,3	76,5	19,1	56,6	75,7	32,8	37,1	69,9
32	7	80,5	102,1	182,6	90,4	97,4	187,8	74,8	69,3	144,1
	14	23,6	81,9	105,5	28	64,7	92,7	41,2	50,6	91,8
	17	20,9	65,9	86,8	27	64,4	91,4	34,8	42,6	77,4
36	8	68,2	118,6	186,8	75,5	105,8	181,3	80,5	72,9	153,4
	16	27,6	97,7	125,3	30,9	70,8	101,7	53,4	54,6	108
	19	23,7	74,5	98,2	24,2	72,9	97,1	38,8	46,4	85,2
40	10	79,4	119,6	487,5	68,9	106,2	175,1	85,2	81,6	166,8
	19	27,8	86,7	114,5	31,1	82,7	113,8	59,2	53,2	112,4
	21	26,2	102,1	128,3	27,8	76,5	104,3	46,8	57,1	103,9

Tabla 8: Tiempos medios de espera, tránsito y total para función de evaluación de tiempo. Interfloor

Interfloor		GA			TS			DC-Elevate		
Nº Plantas	Nº Ascensores	AWT	AJT	ATT	AWT	AJT	ATT	AWT	AJT	ATT
12	2	58	47,5	105,5	67,5	47,4	114,9	48,5	48,7	97,2
	5	14,2	37,8	52	22,8	33,8	56,6	22,7	28	50,7
	7	8,3	30,4	38,7	10,7	29,5	40,2	13,6	21,9	35,5
20	4	55,3	71,7	127	66,9	70,9	137,8	52	52	104
	8	27,4	52,4	79,8	24,1	53,5	77,6	35,9	37,6	73,5
	11	16,2	42,5	58,7	15,9	42,8	58,7	29,1	30,2	59,3
24	5	54,4	80,9	135,3	69,7	82,8	152,5	55,5	55,9	111,4
	10	19,4	59,7	79,1	30,6	55,1	85,7	38,8	42,3	81,1
	13	13,9	49,1	63	12,3	47,4	59,7	26,8	33,1	59,9
28	6	71,9	91,8	163,7	89,9	85,5	175,4	62,7	62,3	125
	13	20	62,4	82,4	26,3	59,3	85,6	45,2	44,2	89,4
	16	14,2	56	70,2	24,1	55,3	79,4	32,8	37,1	69,9
32	7	58,8	91,2	150	94,6	95,7	190,3	77,8	72,3	150,1
	14	22	74,6	96,6	52,3	80,6	132,9	41,2	50,6	91,8
	17	18,4	66	84,4	25,6	70,9	96,5	34,8	42,6	77,4
36	8	79	116,7	195,7	123,8	100,2	224	85,5	75,9	161,4
	16	25,3	79,8	105,1	42,1	91,8	133,9	53,4	54,6	108
	19	17,5	72,6	90,1	18,7	85,6	104,3	38,8	46,4	85,2
40	10	79,1	117,8	196,9	85,5	125,2	210,7	97,2	88,6	185,8
	19	35,5	85,6	121,1	64,1	83,3	147,4	59,2	53,2	112,4
	21	25,4	77,3	102,7	45,8	72,5	118,3	46,8	57,1	103,9

Los resultados anteriores se refieren a los tiempos acontecidos en el edificio. Se muestra a continuación en la Tabla 9 el consumo energético para la función de evaluación de distancia. En la Tabla 10 se expone el consumo energético para la función basada en tiempo.

Tabla 9: Consumo energético para función de evaluación de distancia. Interfloor

Interfloor		GA			TS			DC-Elevate		
Nº Plantas	Nº Ascensores	Consumo Energético (KWh)			Consumo Energético (KWh)			Consumo Energético (KWh)		
12	2									
	5			10,76			10,97			12,28
	7									
20	4									
	8			12,84			12,08			13,58
	11									
24	5									
	10			12,21			13,31			14,14
	13									
28	6									
	13			11,97			14,78			15,45
	16									
32	7									
	14			13,47			15,71			16,29
	17									
36	8									
	16			11,87			14,54			16,37
	19									
40	10									
	19			14,83			16,5			15,55
	21									

Tabla 10: Consumo energético para función de evaluación de tiempo. Interfloor

Interfloor		GA			TS			DC-Elevate		
Nº Plantas	Nº Ascensores	Consumo Energético (KWh)			Consumo Energético (KWh)			Consumo Energético (KWh)		
12	2									
	5			15,18			17,82			18,9
	7									
20	4									
	8			16,62			20,41			20,5
	11									
24	5									
	10			18,12			21,85			21,33
	13									
28	6									
	13			20,95			21,22			23,3
	16									
32	7									
	14			12,81			16,72			24,55
	17									
36	8									
	16			13,82			17,14			16,37
	19									
40	10									
	19			13,91			18,52			15,55
	21									

5.2.2. Tráfico Downpeak

Los resultados obtenidos para el estudio del tráfico downpeak empleando los diferentes métodos de resolución se exponen a continuación. Se muestra en la Tabla 11 los tiempos obtenidos para la función de evaluación basada en distancia y en la Tabla 12 los resultantes para la función basada en tiempo.

Tabla 11: Tiempos medios de espera, tránsito y total para función de evaluación de distancia. Downpeak

Downpeak		GA			TS			DC-Elevate		
Nº Plantas	Nº Ascensores	AWT	AJT	ATT	AWT	AJT	ATT	AWT	AJT	ATT
12	2	49	50,3	99,3	54,2	46,1	100,3	42,2	41,5	83,7
	5	15,7	32,9	48,6	21,3	32,7	54	10,9	24,3	35,2
	7	13	32,3	45,3	19,2	31,2	50,4	8,8	22,9	31,7
20	4	95,6	68,1	163,7	84,3	63	147,3	78,6	64,2	142,8
	8	19,8	51,7	71,5	32,2	16,9	49,1	32,4	42,5	74,9
	11	19	48	67	21,1	39,6	60,7	21,4	38,6	60
24	5	134,2	72,8	207	102,7	70,7	173,4	68,2	47,9	116,1
	10	30,7	49,9	80,6	34,4	51,7	86,1	35,8	35,9	71,7
	13	28,7	51	79,7	25,9	45,6	71,5	26,6	33,9	60,5
28	6	109,8	88,5	198,3	120,3	83,3	203,6	65,1	51,3	116,4
	13	27,4	61,9	89,3	32,3	56,7	89	28,2	39	67,2
	16	22,4	57,8	80,2	25,7	51,8	77,5	21,1	37	58,1
32	7	125,6	90,5	216,1	106,4	84,2	190,6	125,2	59,6	184,8
	14	30,4	66,7	97,1	38,7	63	101,7	36,7	54	90,7
	17	23,9	62,7	86,6	26,2	54,7	80,9	28,3	45	73,3
36	8	115,5	85,2	200,7	101,5	75,2	176,7	116,7	64,3	181
	16	30	74,1	104,1	55,3	66,3	121,6	55,7	62,1	117,8
	19	29,4	69,8	99,2	40,2	61,2	101,4	32,4	44,4	76,8
40	10	85,6	78,1	163,7	70,1	71,2	141,3	100,1	115,4	215,5
	19	36,7	77,1	113,8	53,7	71,4	125,1	62,9	67	129,9
	21	28,9	75,1	104	39,4	72,9	112,3	41,9	64,6	106,5

Tabla 12: Tiempos medios de espera, tránsito y total para función de evaluación de tiempo. Downpeak

Downpeak		GA			TS			DC-Elevate		
Nº Plantas	Nº Ascensores	AWT	AJT	ATT	AWT	AJT	ATT	AWT	AJT	ATT
12	2	57,9	45,3	103,2	65,4	42,4	107,8	42,2	41,5	83,7
	5	18,7	34,2	52,9	27,5	31,8	59,3	10,9	24,3	35,2
	7	11,9	27,7	39,6	24	30,7	54,7	8,8	22,9	31,7
20	4	51,3	67,2	118,5	87	64,2	151,2	78,6	64,2	142,8
	8	22	52,3	74,3	41,3	48,5	89,8	32,4	42,5	74,9
	11	15,8	44,5	60,3	33,3	42,6	75,9	21,4	38,6	60
24	5	85,1	71,6	156,7	122,7	72,3	195	68,2	47,9	116,1
	10	22,4	60,4	82,8	52,7	60,3	113	35,8	35,9	71,7
	13	18,5	56,1	74,6	48,8	55	103,8	26,6	33,9	60,5
28	6	64,4	80,5	144,9	95,8	80,2	176	65,1	51,3	116,4
	13	26,9	60,5	87,4	77	70	147	28,2	39	67,2
	16	18,1	51,9	70	61,5	56,4	117,9	21,1	37	58,1
32	7	97,7	93,4	191,1	138,5	85,4	223,9	154	60	214
	14	26,6	71,2	97,8	98	82,6	180,6	36,7	54	90,7
	17	22,7	62	84,7	70,5	82,1	152,6	28,3	45	73,3
36	8	110,8	101,7	212,5	150,5	94	244,5	116,7	64,3	181
	16	29,9	80	109,9	75,4	90,8	166,2	55,7	62,1	117,8
	19	27,4	72,2	99,6	58,3	85,7	144	32,4	44,4	76,8
40	10	95,9	108,6	204,5	135,5	100,5	236	325,8	115,4	441,2
	19	30	88,9	118,9	85,6	92,3	177,9	62,9	67	129,9
	21	24,2	77,8	102	65,4	95,8	161,2	41,9	64,6	106,5

En las Tabla 13 y Tabla 14 se muestran el consumo energético obtenido tras las simulaciones para la función de evaluación basada en distancia y tiempo, respectivamente.

Tabla 13: Consumo energético para función de evaluación de distancia. Downpeak

Downpeak		GA			TS			DC-Elevate		
Nº Plantas	Nº Ascensores	Consumo Energético (KWh)			Consumo Energético (KWh)			Consumo Energético (KWh)		
12	2									
	5		9,76			12,1			11,4	
	7									
20	4									
	8		14,86			15,13			19,64	
	11									
24	5									
	10		16,06			16,6			21,48	
	13									
28	6									
	13		17,78			17,22			23,2	
	16									
32	7									
	14		16,36			18,13			23,28	
	17									
36	8									
	16		20,34			22,48			26,94	
	19									
40	10									
	19		16,15			20,18			24,35	
	21									

Tabla 14: Consumo energético para función de evaluación de tiempo. Downpeak

Downpeak		GA			TS			DC-Elevate		
Nº Plantas	Nº Ascensores	Consumo Energético (KWh)			Consumo Energético (KWh)			Consumo Energético (KWh)		
12	2									
	5		13,56			8,08			17,18	
	7									
20	4									
	8		16,05			10,59			19,64	
	11									
24	5									
	10		15,8			13,95			19,03	
	13									
28	6									
	13		15,88			17,44			23,21	
	16									
32	7									
	14		19,71			21,14			23,28	
	17									
36	8									
	16		19,16			19,53			26,94	
	19									
40	10									
	19		19,7			21,44			24,35	
	21									

5.2.3. Tráfico Lunchpeak

El estudio del tráfico lunchpeak empleando los diferentes métodos de resolución arroja los siguientes resultados. Se muestra en la Tabla 15 los tiempos obtenidos para la función de evaluación basada en distancia y en la Tabla 16 los resultantes para la función basada en tiempo.

Tabla 15: Tiempos medios de espera, tránsito y total para función de evaluación de distancia. Lunchpeak

Lunchpeak		GA				TS				DC-Elevate			
Nº Plantas	Nº Ascensores	AWT	AJT	ATT	LWT	AWT	AJT	ATT	LWT	AWT	AJT	ATT	LWT
12	2	40,5	46,4	86,9	155,4	43	45,4	88,4	327	40,1	40,2	80,3	137,6
	5	9,5	30,5	40	108,7	13,6	30,9	44,5	195,1	11	23,6	34,6	51,6
	7	7	28,5	35,5	104,2	8,5	27,2	35,7	161,5	9,5	22,3	31,8	33,6
20	4	40,1	70,3	110,4	279,6	48,1	69,5	117,6	332,7	43,9	46,4	90,3	213,7
	8	15	47,5	62,5	204,5	17,2	44,7	61,9	152,7	20,3	36,3	56,6	113,3
	11	8,6	40,7	49,3	211,5	12,8	39	51,8	132	15,7	29	44,7	43,5
24	5	37,5	81,5	119	315,2	55,5	76	131,5	286,5	48,3	49,5	97,8	194
	10	15,1	52,7	67,8	178,8	18,6	51,3	69,9	187,3	18,3	35,6	53,9	92
	13	9,8	46,6	56,4	169,3	13,7	41,8	55,5	161,5	15,1	32,6	47,7	56
28	6	49,6	90,4	140	448,5	63,2	87,1	150,3	732	52,6	53,1	105,7	256,1
	13	13,3	55	68,3	250	19	54,1	73,1	247,6	21,9	37,7	59,6	137,9
	16	10,7	49,3	60	196,5	15,3	50,2	65,5	295,4	15,9	35,8	51,7	99
32	7	70,5	100,7	171,2	818,9	84,5	90,4	174,9	358,4	71,1	65,8	136,9	331,9
	14	16,8	65,2	82	156,3	23,5	63,1	86,6	284,7	25,5	52,9	78,4	136,6
	17	13,2	54,2	67,4	105	18,9	56,6	75,5	115,6	15,1	44,2	59,3	101,5
36	8	45,8	94,2	140	358,9	55,4	84,6	140	524,9	69,2	65,5	134,7	325
	16	17,4	69,8	87,2	257,7	21,3	67,6	88,9	345,2	25,3	55,6	80,9	184,8
	19	13,6	65	78,6	252,6	18,8	64,4	83,2	325,4	22,7	51,5	74,2	125,5
40	10	35,4	82,6	118	358,7	45,2	71,6	116,8	456,5	86,4	78,1	164,5	727,6
	19	21,5	73,4	94,9	274,5	26,8	73,2	100	285,4	42,3	55,5	97,8	169,3
	21	13,3	70,4	83,7	302,6	24,4	73	97,4	356,8	30,4	50,3	80,7	127,6

Tabla 16: Tiempos medios de espera, tránsito y total para función de evaluación de tiempo. Lunchpeak

Lunchpeak		GA				TS				DC-Elevate			
Nº Plantas	Nº Ascensores	AWT	AJT	ATT	LWT	AWT	AJT	ATT	LWT	AWT	AJT	ATT	LWT
12	2	39,5	47	86,5	155,4	45	45,9	90,9	327	40,1	40,2	80,3	137,6
	5	9,4	30,8	40,2	108,7	18,5	31	49,5	195,1	11	23,6	34,6	51,6
	7	6,1	28,2	34,3	104,2	11,7	29,4	41,1	161,5	9,5	22,3	31,8	33,6
20	4	44,4	70,8	115,2	279,6	53	66,7	119,7	332,7	43,9	46,4	90,3	213,7
	8	14,6	46,5	61,1	204,5	27,9	45,6	73,5	152,7	20,3	36,3	56,6	113,3
	11	10,2	41,4	51,6	211,5	25,8	44,5	70,3	132	15,7	29	44,7	43,5
24	5	40,4	80,4	120,8	315,2	62,7	74,3	137	286,5	48,3	49,5	97,8	194
	10	14,2	53,7	67,9	178,8	30,3	51,4	81,7	187,3	18,3	35,6	53,9	92
	13	10,8	52,4	63,2	169,3	21,9	48,6	70,5	161,5	15,1	32,6	47,7	56
28	6	56	87,1	143,1	448,5	74,7	81,2	155,9	732	52,6	53,1	105,7	256,1
	13	15,9	51,4	67,3	250	38,7	59,5	98,2	247,6	21,9	37,7	59,6	137,9
	16	12,5	45,6	58,1	196,5	29,7	55,8	85,5	295,4	15,9	35,8	51,7	99
32	7	35,6	70,1	105,7	1806,9	60,2	80,1	140,3	1256	75,1	67,8	142,9	331,9
	14	19,5	48,7	68,2	156,3	55,8	71,6	127,4	284,7	28,5	45,9	74,4	136,6
	17	14,2	49,1	63,3	105	48,3	68,4	116,7	115,6	17,1	48,2	65,3	101,5
36	8	30,5	62	92,5	1567	59,1	80,5	139,6	1356,9	75,2	68,5	143,7	325
	16	15,6	47,6	63,2	257,7	48,9	75,6	124,5	345,2	29,3	54,6	83,9	184,8
	19	13,2	49,8	63	252,6	40,6	72,2	112,8	325,4	22,7	51,5	74,2	125,5
40	10	42,6	95	137,6	3111,6	80,5	90,1	170,6	2596	89,4	78,1	167,5	727,6
	19	18,1	69,2	87,3	274,5	65,8	85,6	151,4	285,4	45,3	50	95,3	169,3
	21	14,6	56	70,6	302,6	53,4	68,4	121,8	356,8	31,4	48,3	79,7	127,6

El consumo energético que se ha obtenido en la simulación se muestra en la Tabla 17 y Tabla 18 en función de si su evaluación se ha realizado mediante distancia o tiempo.

Tabla 17: Consumo energético para función de evaluación de distancia. Lunchpeak

Lunchpeak		GA	TS	DC-Elevate
Nº Plantas	Nº Ascensores	Consumo Energético (KWh)	Consumo Energético (KWh)	Consumo Energético (KWh)
12	2			
	5	13,63	12,1	17,15
	7			
20	4			
	8	19,39	17,11	20,05
	11			
24	5			
	10	18,71	17,37	21,59
	13			
28	6			
	13	18,88	20,24	23,72
	16			
32	7			
	14	20,89	21,57	23,55
	17			
36	8			
	16	21,74	21,24	26,04
	19			
40	10			
	19	18,96	23,05	25,44
	21			

Tabla 18: Consumo energético para función de evaluación de tiempo. Lunchpeak

Lunchpeak		GA	TS	DC-Elevate
Nº Plantas	Nº Ascensores	Consumo Energético (KWh)	Consumo Energético (KWh)	Consumo Energético (KWh)
12	2			
	5	11,59	11,45	15,24
	7			
20	4			
	8	18,01	18,06	20,05
	11			
24	5			
	10	20,95	18,2	21,59
	13			
28	6			
	13	19,07	22,68	23,72
	16			
32	7			
	14	18,51	21,6	23,55
	17			
36	8			
	16	19,94	24,46	26,04
	19			
40	10			
	19	19,89	21,08	25,44
	21			

6. ANÁLISIS COMPARATIVO DE TIEMPOS DE ESPERA

El presente apartado se emplea para realizar el estudio comparativo de las simulaciones que se han llevado a cabo mediante el software *Elevate* tras su programación mediante *Visual Studio*. Se trata de analizar para cada patrón de tráfico la cantidad de cabinas que se tiene para cada edificio con un determinado número de plantas. La comparación se realiza observando la diferenciación que se produce al emplear diferentes funciones de evaluación, así como diferentes métodos de resolución al problema de asignación de cabinas a llamadas.

Se lleva a cabo en primer lugar el análisis de situaciones basadas en el estudio de tiempos en el edificio. Es decir, se estudia el tiempo medio de espera, tránsito y total en función del método de resolución y de la función de evaluación empleadas.

6.1. Función de evaluación basada en distancia

Para cada tipo de tráfico se analiza el comportamiento del tráfico vertical en función de la cantidad de plantas y ascensores que dispone el edificio, así como del método de resolución.

6.1.1. Tráfico Interfloor

Edificio de 12 plantas

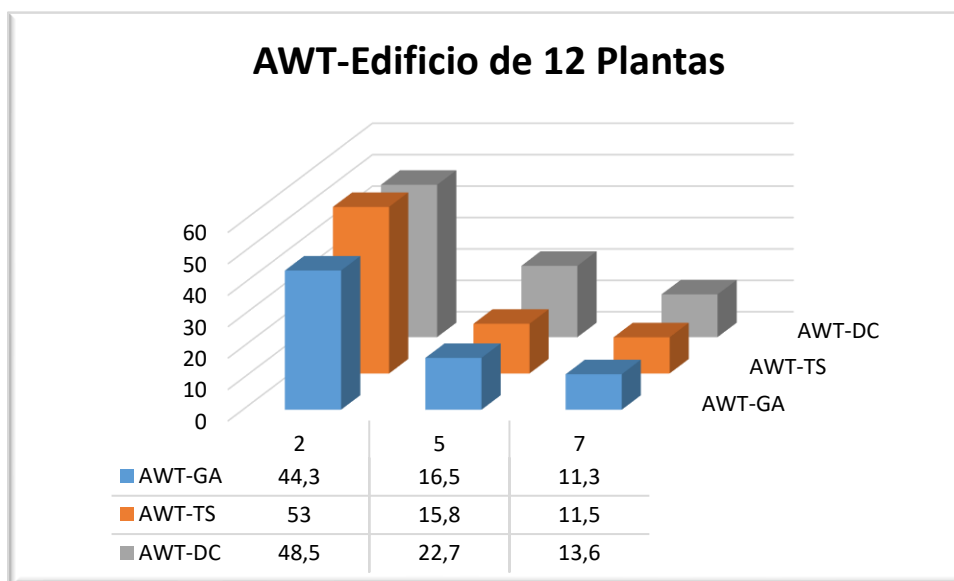


Figura 31: AWT (función de evaluación: distancia)- 12 Plantas. Interfloor

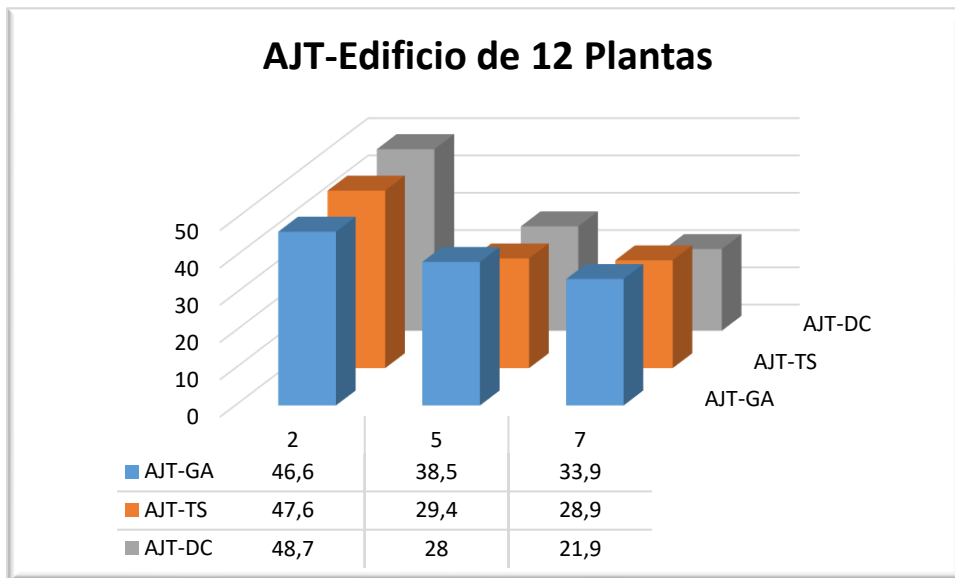


Figura 32: AJT (función de evaluación: distancia)- 12 Plantas. Interfloor

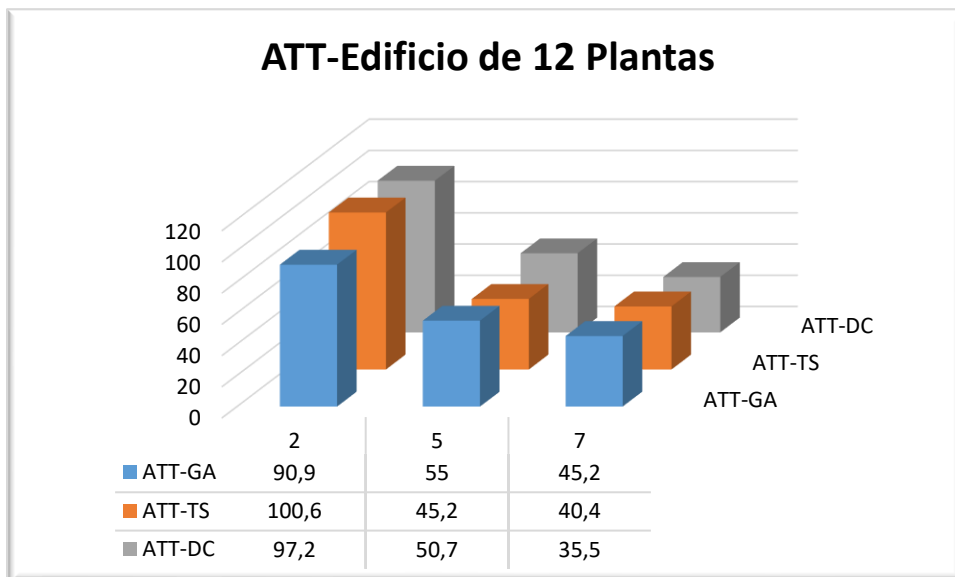


Figura 33: ATT (función de evaluación: distancia)- 12 Plantas. Interfloor

Para un edificio de 12 plantas se observa en la Figura 31 que, para la misma cantidad de ascensores, los tres algoritmos presentan un tiempo medio de espera parecido, aunque mediante el algoritmo genético y la búsqueda tabú se obtienen mejores que con el algoritmo implementado por *Elevate*. En cambio, para el tiempo medio de tránsito se observa en la Figura 32 que el algoritmo DC-Elevate obtiene los mejores tiempos, seguido por la búsqueda tabú. Se obtiene finalmente que el tiempo medio total presenta al algoritmo genético y a la búsqueda tabú como mejores opciones según la Figura 33.

Edificio de 20 plantas

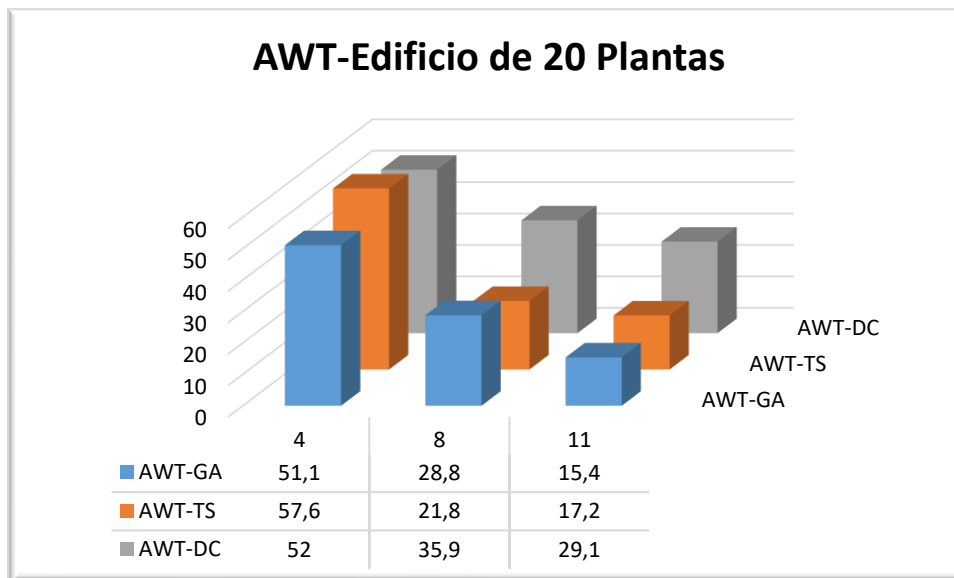


Figura 34: AWT (función de evaluación: distancia)- 20 Plantas. Interfloor

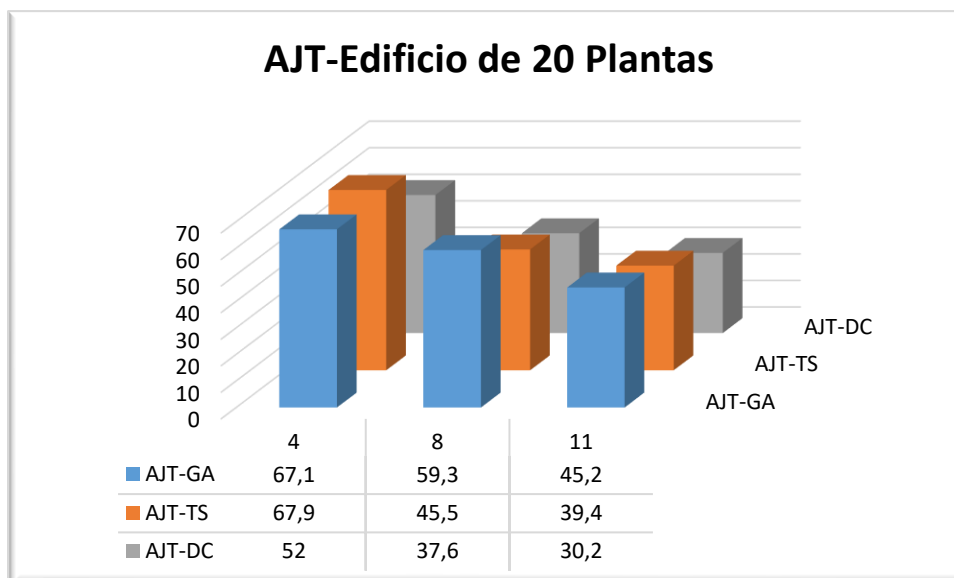


Figura 35: AJT (función de evaluación: distancia)- 20 Plantas. Interfloor

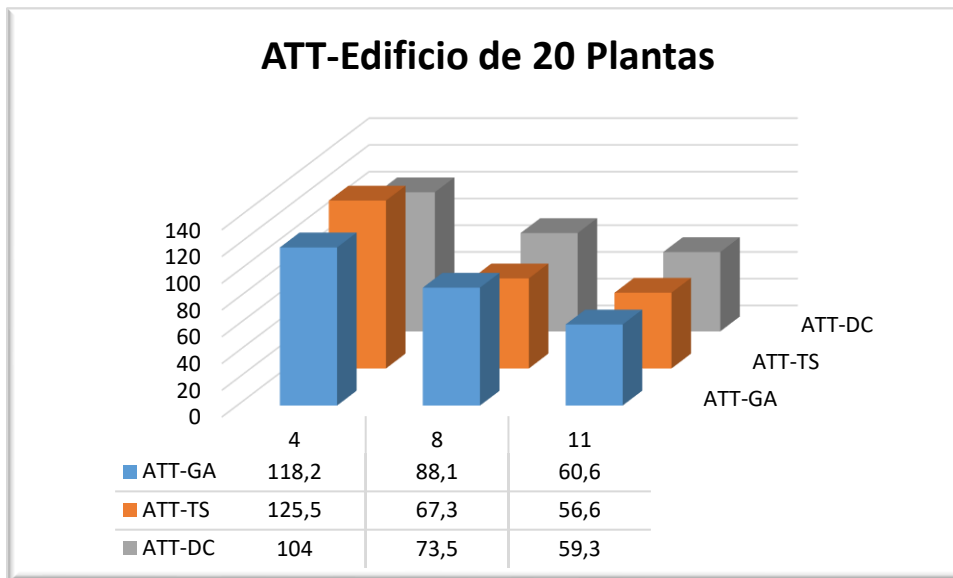


Figura 36: ATT (función de evaluación: distancia)- 20 Plantas. Interfloor

Para un edificio de 20 plantas se observa en la Figura 34 como el tiempo medio de espera es par al emplear el algoritmo genético y la búsqueda tabú frente al algoritmo DC-Elevate. El tiempo medio de tránsito se consigue en su mejor versión para el algoritmo implementado por *Elevate* mientras que el tiempo medio total obtiene mejores resultados para la búsqueda tabú tal y como se aprecia en la Figura 36.

Edificio de 24 plantas

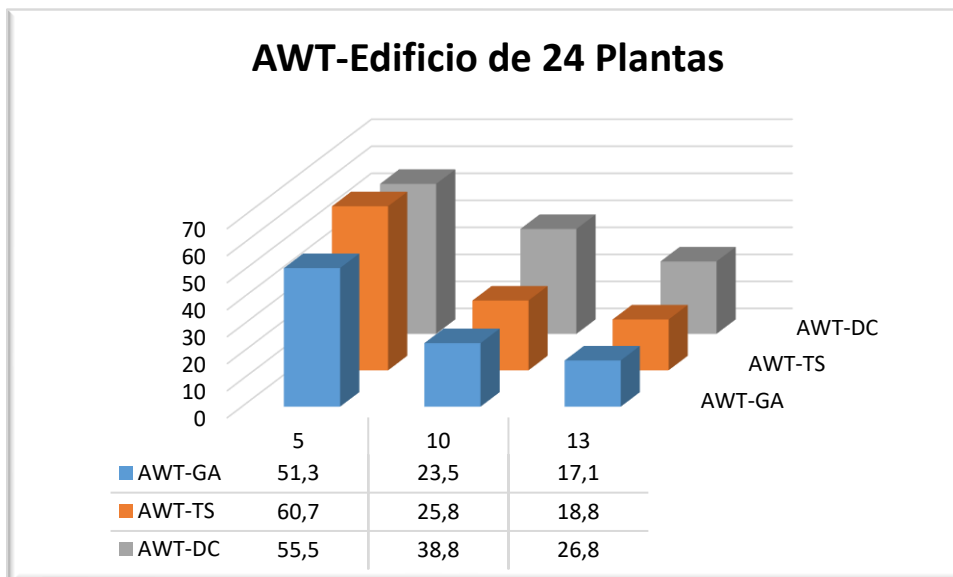


Figura 37: AWT (función de evaluación: distancia)- 24 Plantas. Interfloor

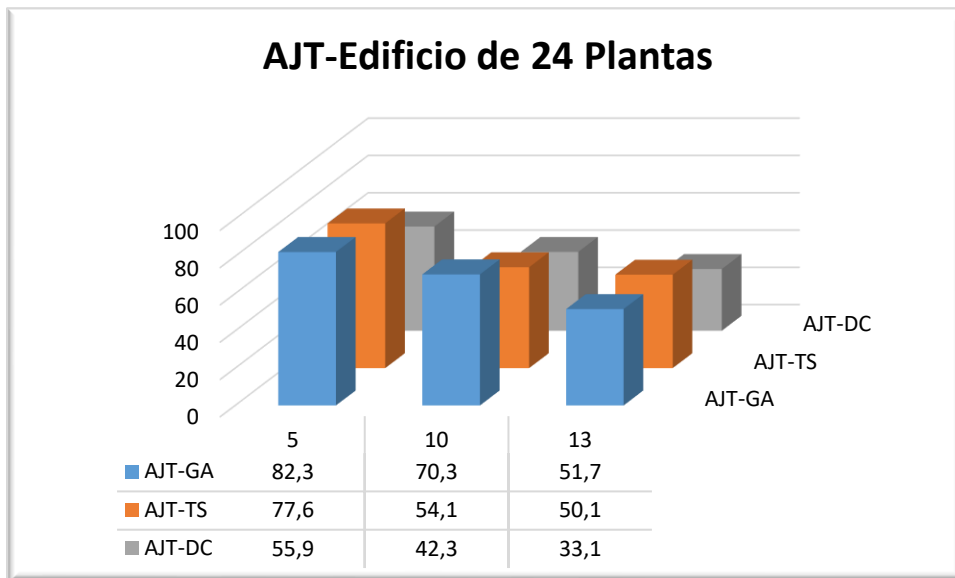


Figura 38: AJT (función de evaluación: distancia)- 24 Plantas. Interfloor

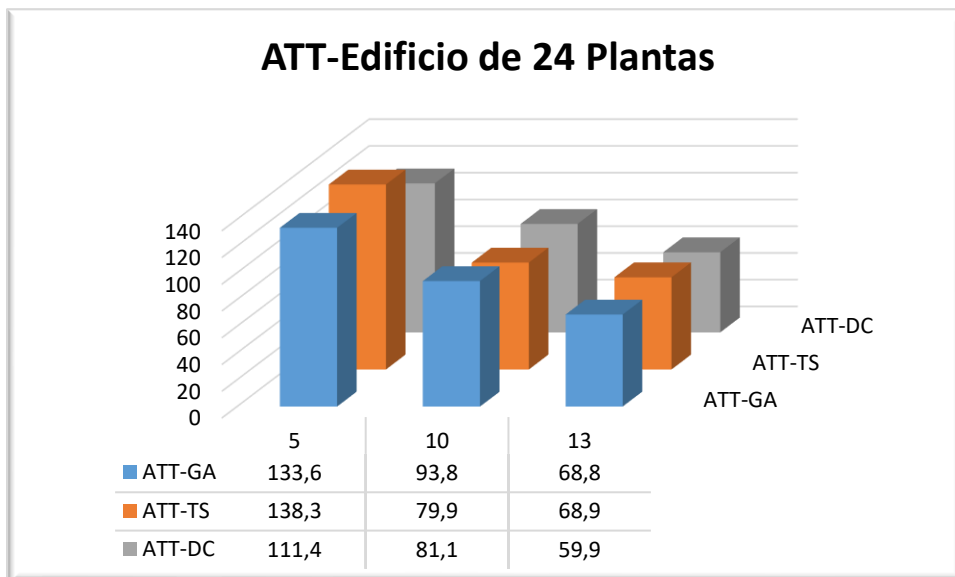


Figura 39: ATT (función de evaluación: distancia)- 24 Plantas. Interfloor

Se observa en la Figura 37 que el algoritmo genético obtiene los mejores tiempos medios de espera mientras que el tiempo medio de tránsito obtiene los mejores valores empleando DC-Elevate seguido del algoritmo de búsqueda tabú. La Figura 39 muestra como los mejores valores los obtienen tanto la búsqueda tabú como el DC-Elevate.

Edificio de 28 plantas

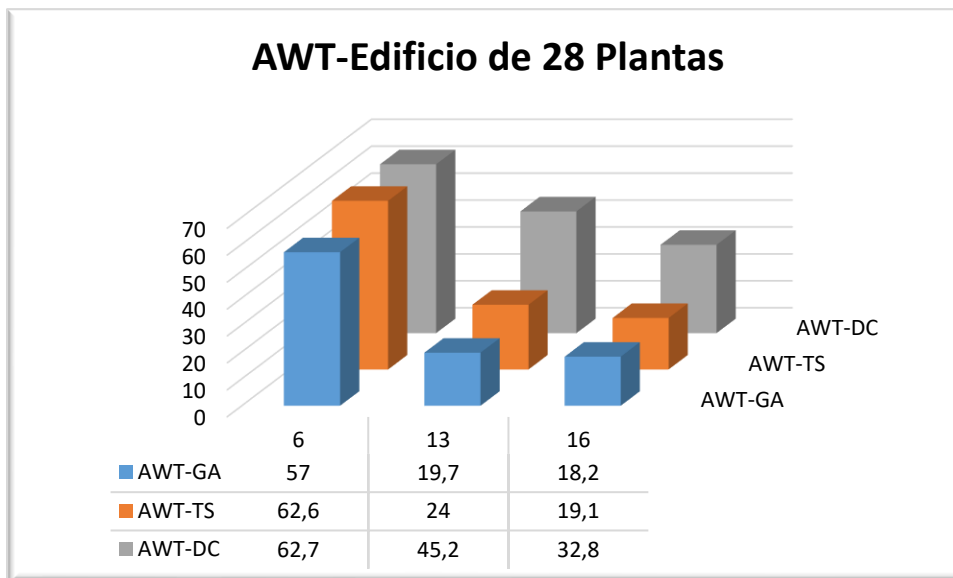


Figura 40: AWT (función de evaluación: distancia)- 28 Plantas. Interfloor

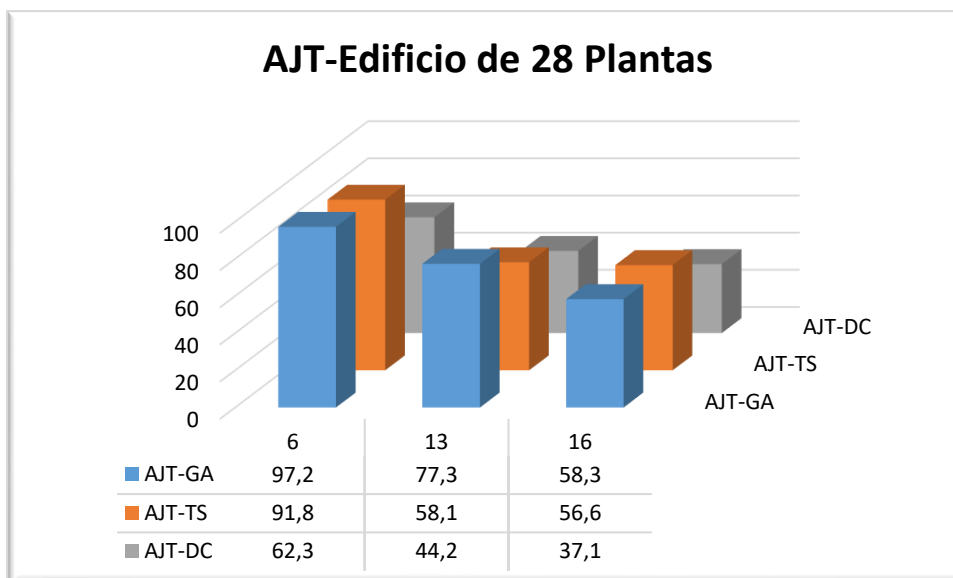


Figura 41: AJT (función de evaluación: distancia)- 28 Plantas. Interfloor

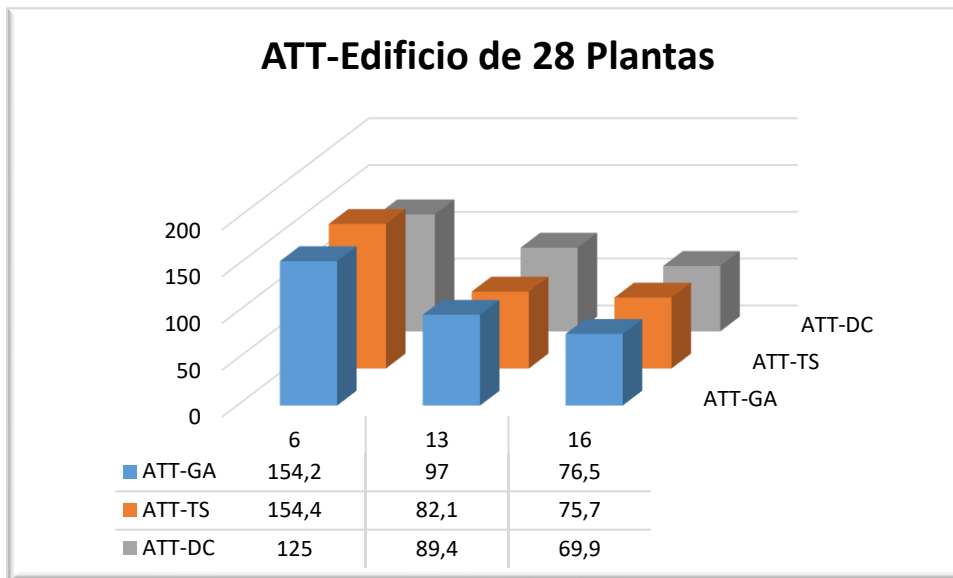


Figura 42: ATT (función de evaluación: distancia)- 28 Plantas. Interfloor

Para un edificio de 28 plantas se observa que el tiempo medio de espera de los pasajeros obtiene los mejores valores empleando el algoritmo genético seguido muy de cerca de la búsqueda tabú. Para el tiempo medio de tránsito se observa en la Figura 41 como DC-Elevate obtiene los mejores valores. En cambio, el tiempo total medio presenta los mejores resultados en forma de empate entre los tres algoritmos.

Edificio de 32 plantas

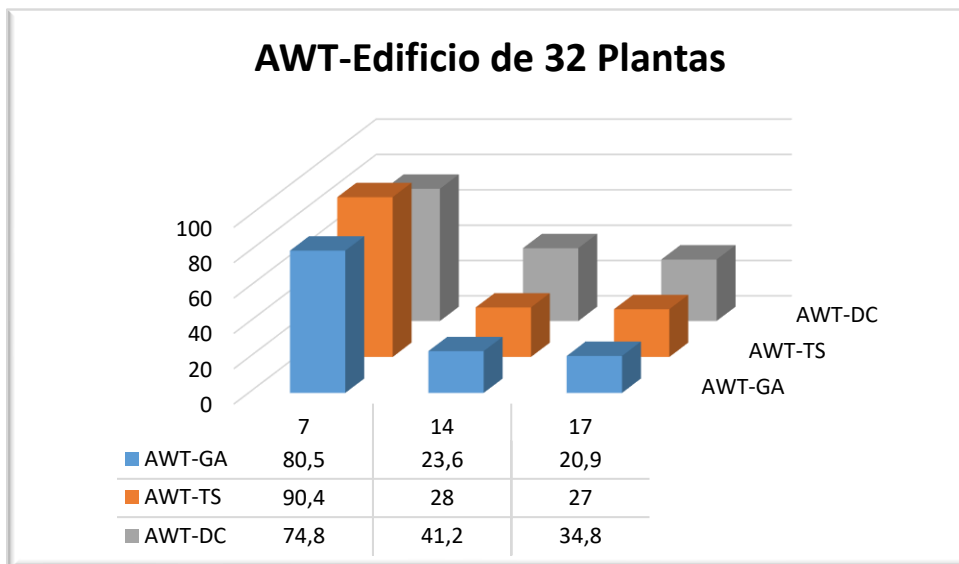


Figura 43: AWT (función de evaluación: distancia)- 32 Plantas. Interfloor

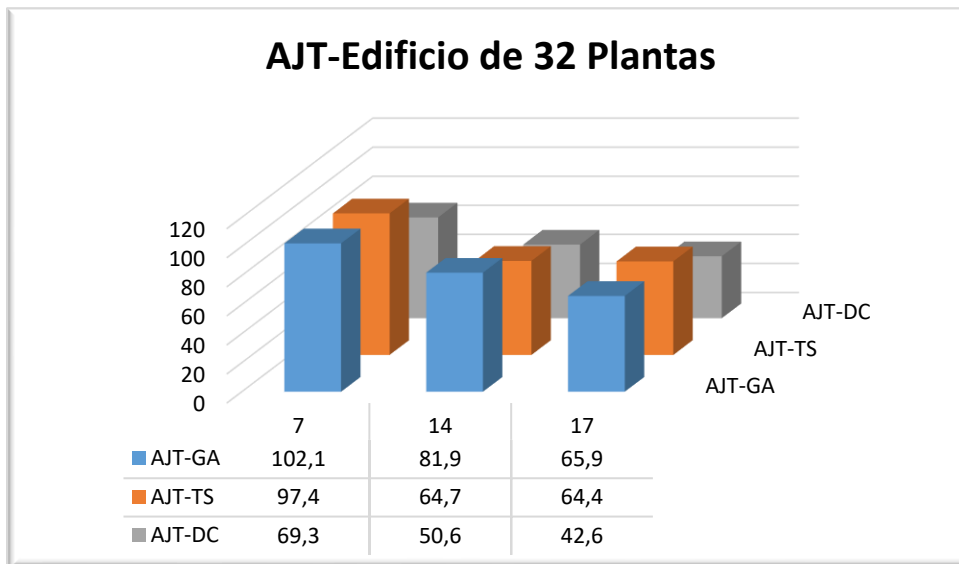


Figura 44: AJT (función de evaluación: distancia)- 32 Plantas. Interfloor

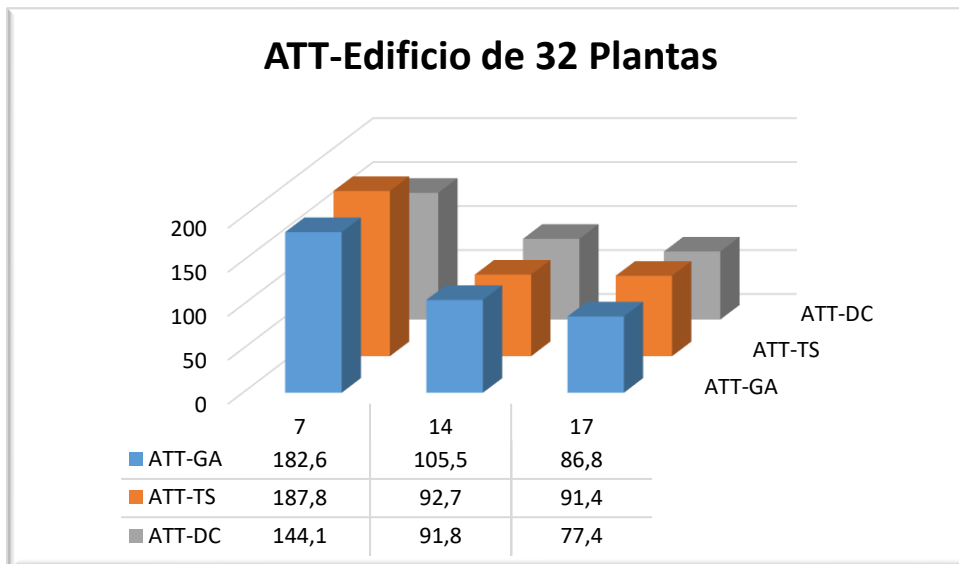


Figura 45 ATT (función de evaluación: distancia)- 32 Plantas. Interfloor

La Figura 43 presenta los mejores resultados para la programación mediante el algoritmo genético mientras que el algoritmo implementado por *Elevate* ofrece los mejores tiempos medios de tránsito. Para el tiempo medio total se tiene que los tres algoritmos ofrecen buenos resultados.

Edificio de 36 plantas

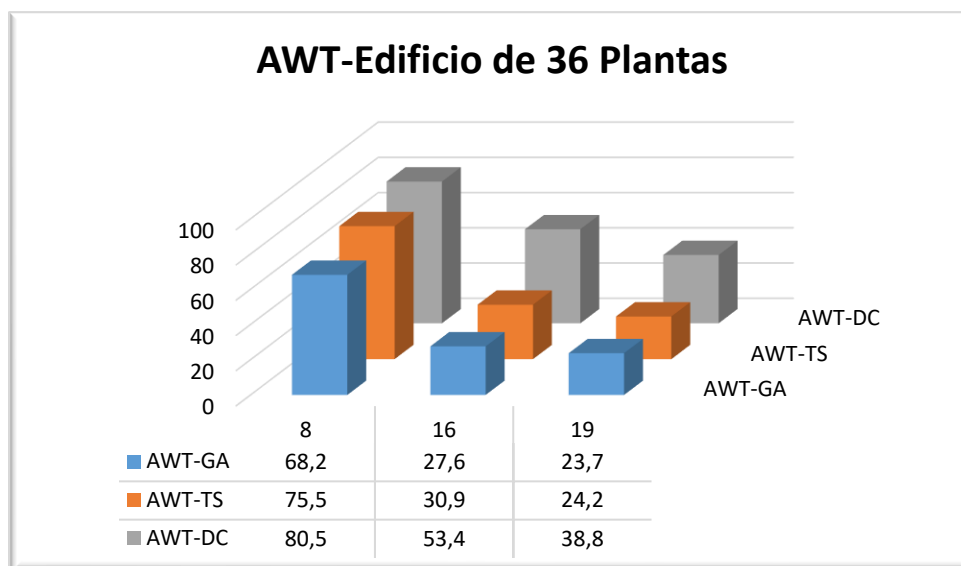


Figura 46: AWT (función de evaluación: distancia)- 36 Plantas. Interfloor

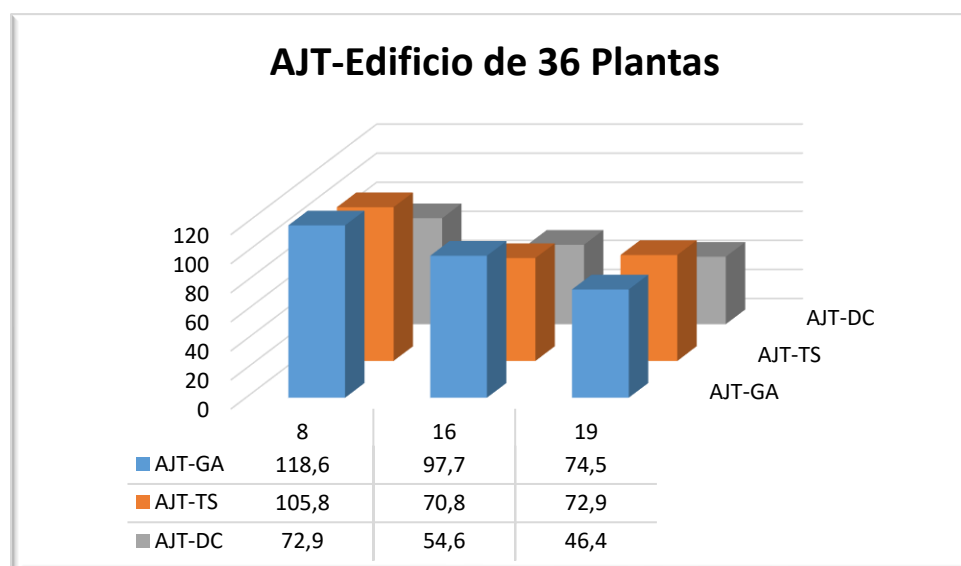


Figura 47: AJT (función de evaluación: distancia)- 36 Plantas. Interfloor

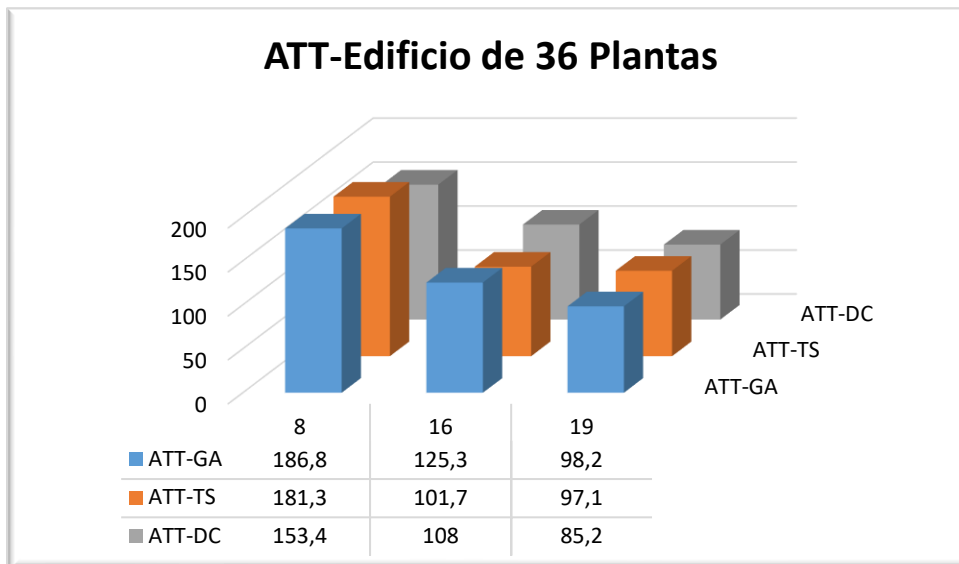


Figura 48: ATT (función de evaluación: distancia)- 36 Plantas. Interfloor

Para un edificio de 36 plantas se muestra en la Figura 46 que el mejor AWT resulta del empleo del algoritmo genético o la búsqueda tabú. El tiempo medio de tránsito mejor es obtenido mediante DC-Elevate frente al tiempo medio total cuyo mejor valor lo obtiene el algoritmo de búsqueda tabú según Figura 48.

Edificio de 40 plantas

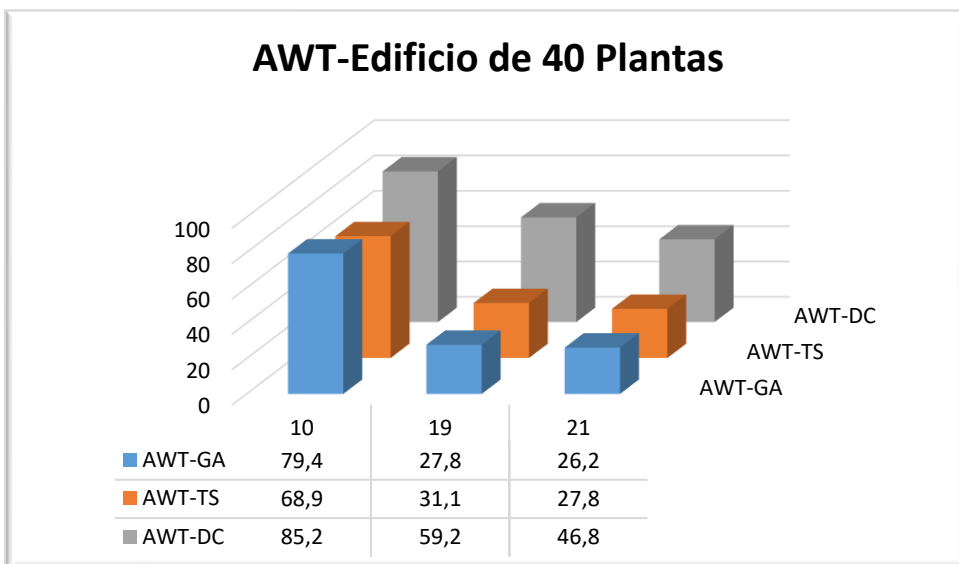


Figura 49: AWT (función de evaluación: distancia)- 40 Plantas. Interfloor

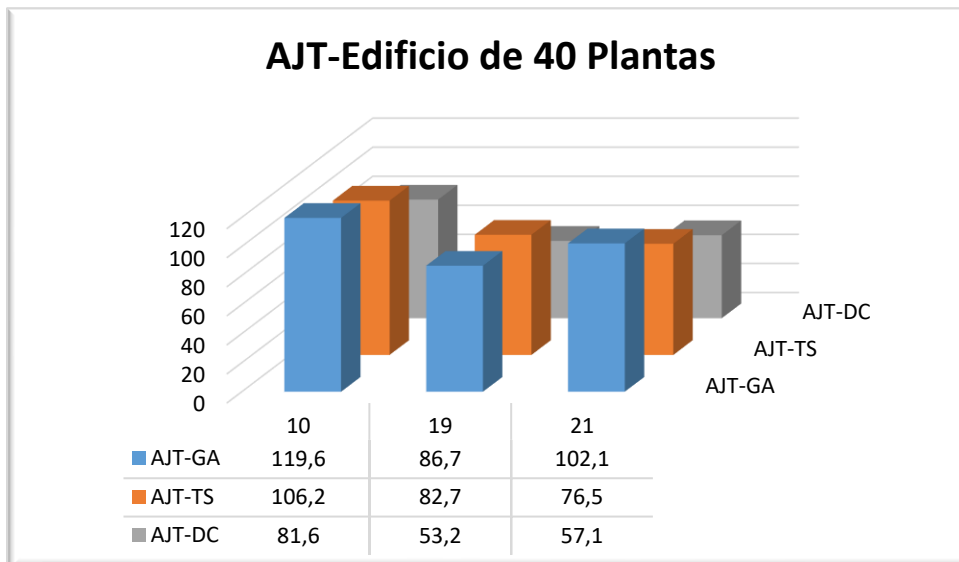


Figura 50: AJT (función de evaluación: distancia)- 40 Plantas. Interfloor

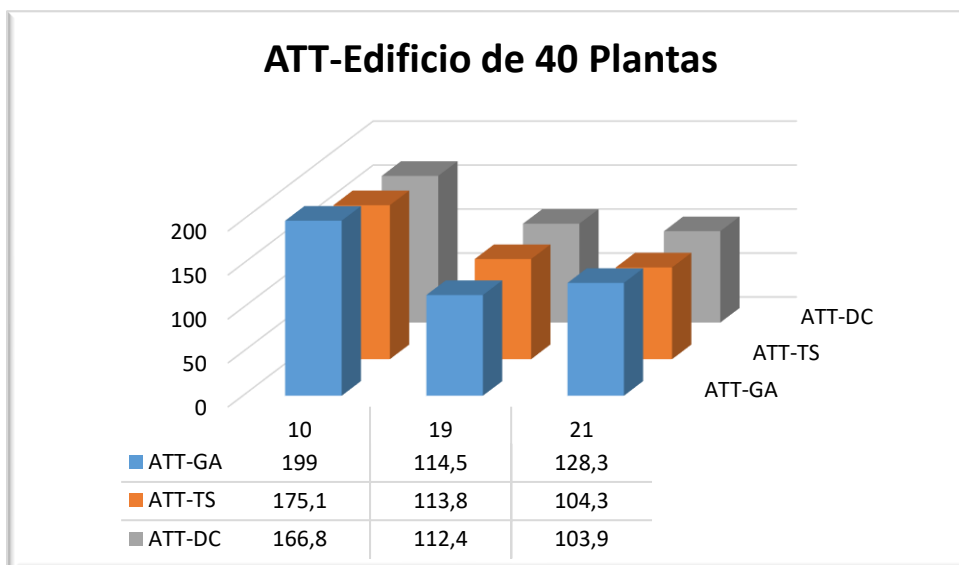


Figura 51: ATT (función de evaluación: distancia)- 40 Plantas. Interfloor

Para un edificio de 40 plantas se muestra que el tiempo medio de espera obtiene los mejores valores para la programación mediante la búsqueda tabú y algoritmo genético. La Figura 50 muestra como el AJT obtiene los mejores resultados para DC-Elevate. En cambio, los mejores resultados se obtienen para el tiempo medio total mediante la programación del algoritmo de búsqueda tabú y DC-Elevate.

6.1.2. Tráfico Lunchpeak

Edificio de 12 plantas

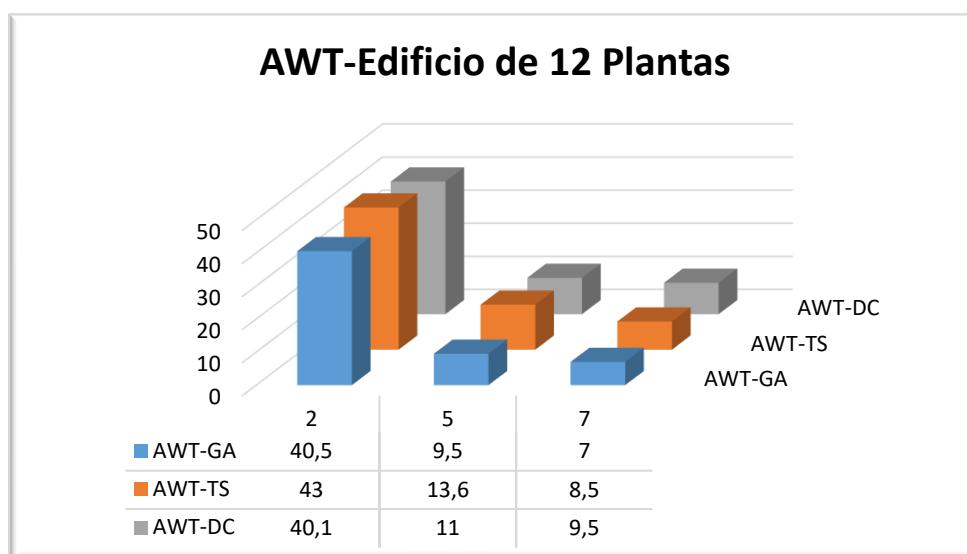


Figura 52: AWT (función de evaluación: distancia)- 12 Plantas. Lunchpeak

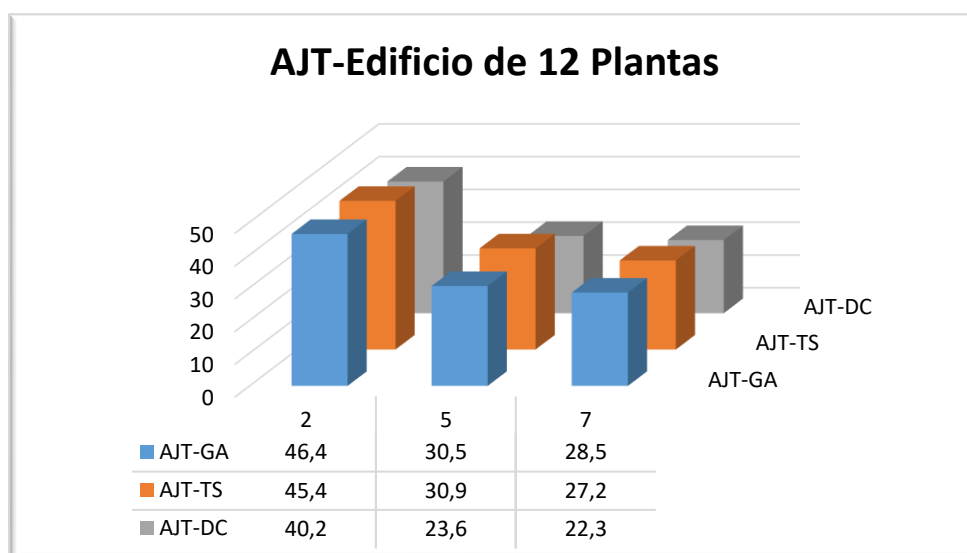


Figura 53: AJT (función de evaluación: distancia)- 12 Plantas. Lunchpeak

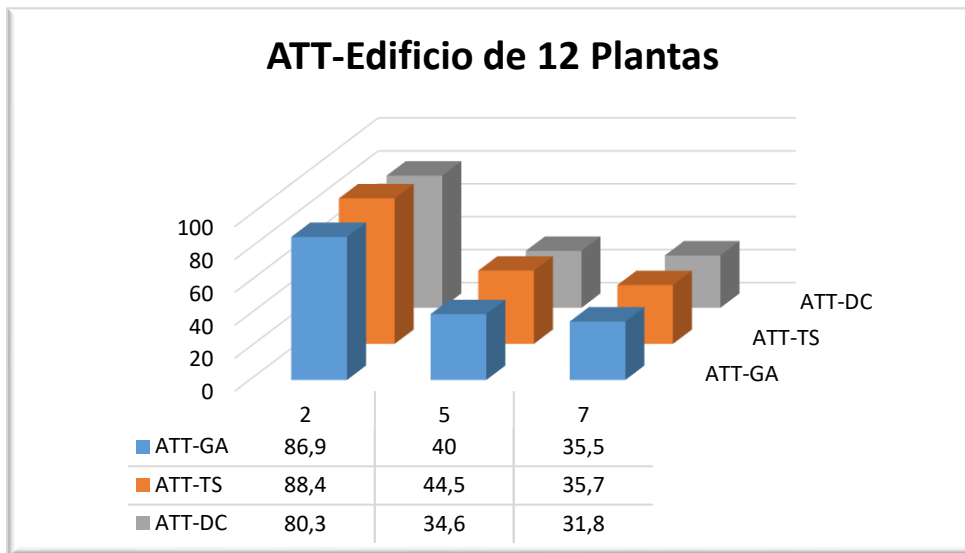


Figura 54: ATT (función de evaluación: distancia)- 12 Plantas. Lunchpeak

Para un edificio de 12 plantas se observa en la Figura 52 que el algoritmo genético presenta los mejores tiempos medios de espera mientras que para el tiempo medio de tránsito lo obtiene el algoritmo implementado en *Elevate*. Para el tiempo medio total se obtienen los mejores tiempos mediante el algoritmo DC-Elevate, pero seguido muy de cerca por los otros dos algoritmos.

Edificio de 20 plantas

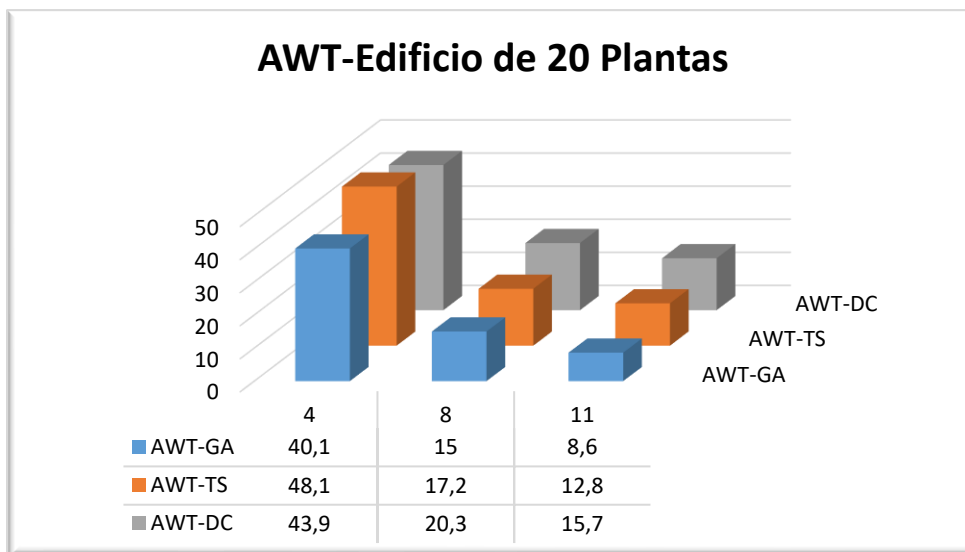


Figura 55: AWT (función de evaluación: distancia)- 20 Plantas. Lunchpeak

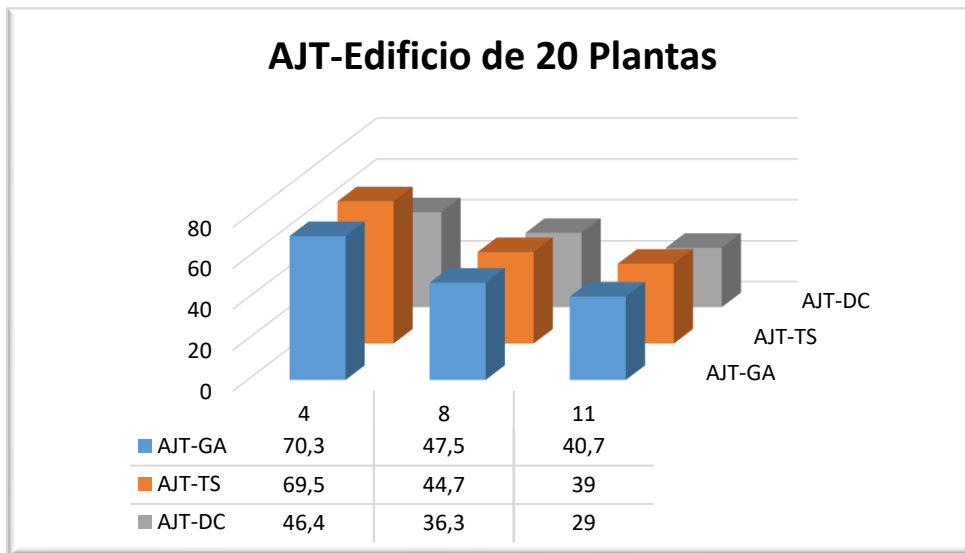


Figura 56: AJT (función de evaluación: distancia)- 20 Plantas. Lunchpeak

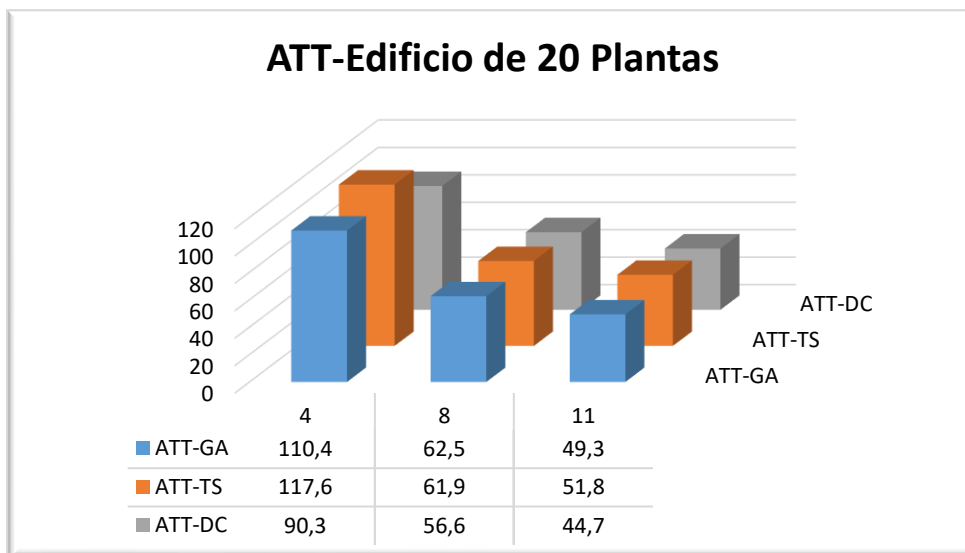


Figura 57: ATT (función de evaluación: distancia)- 20 Plantas. Lunchpeak

En un edificio de 20 plantas se tiene que el tiempo medio de espera más favorable lo obtiene el algoritmo genético. Se muestra en la Figura 56 como el tiempo medio de tránsito con una mayor bondad se obtiene mediante el empleo del algoritmo de búsqueda tabú y el algoritmo implementado por *Elevate*. La Figura 57 presenta al algoritmo genético como la mejor opción para el tiempo medio total.

Edificio de 24 plantas

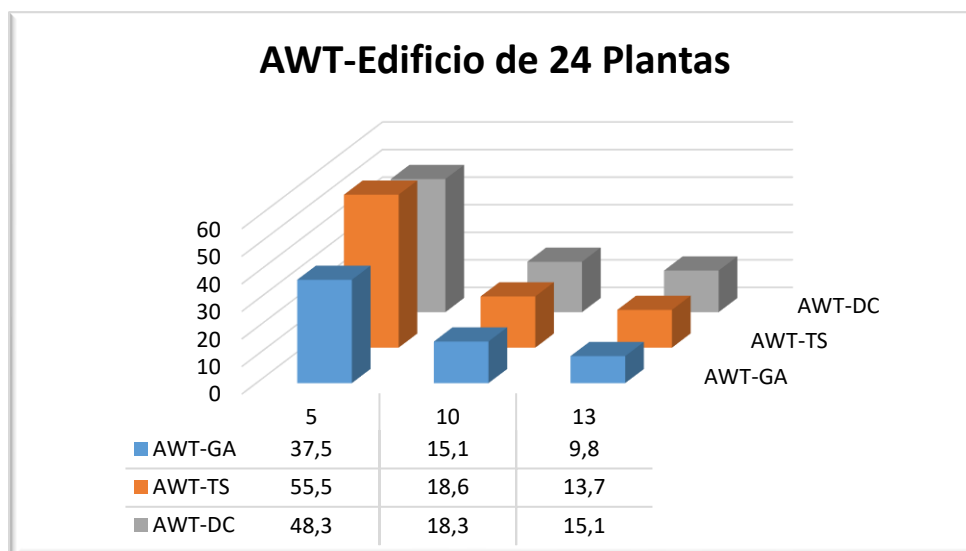


Figura 58: AWT (función de evaluación: distancia)- 24 Plantas. Lunchpeak

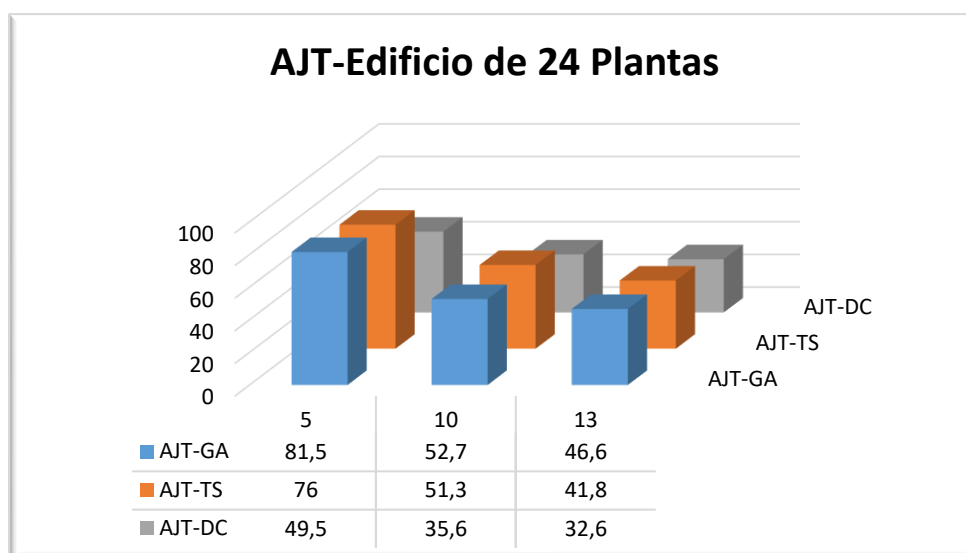


Figura 59: AJT (función de evaluación: distancia)- 24 Plantas. Lunchpeak

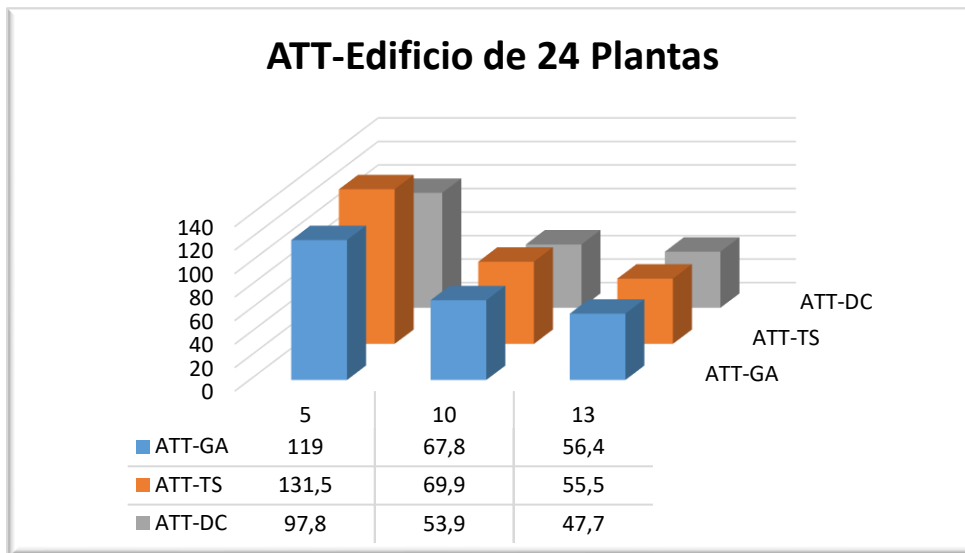


Figura 60: ATT (función de evaluación: distancia)- 24 Plantas. Lunchpeak

El algoritmo genético presenta los mejores resultados para el tiempo medio de espera en un edificio de 24 plantas mientras que el algoritmo de búsqueda tabú sigue al algoritmo DC-Elevate como mejor propuesta para el tiempo medio de tránsito según la Figura 59. Para el tiempo medio total se observa que los tres algoritmos presentan valores parecidos.

Edificio de 28 plantas

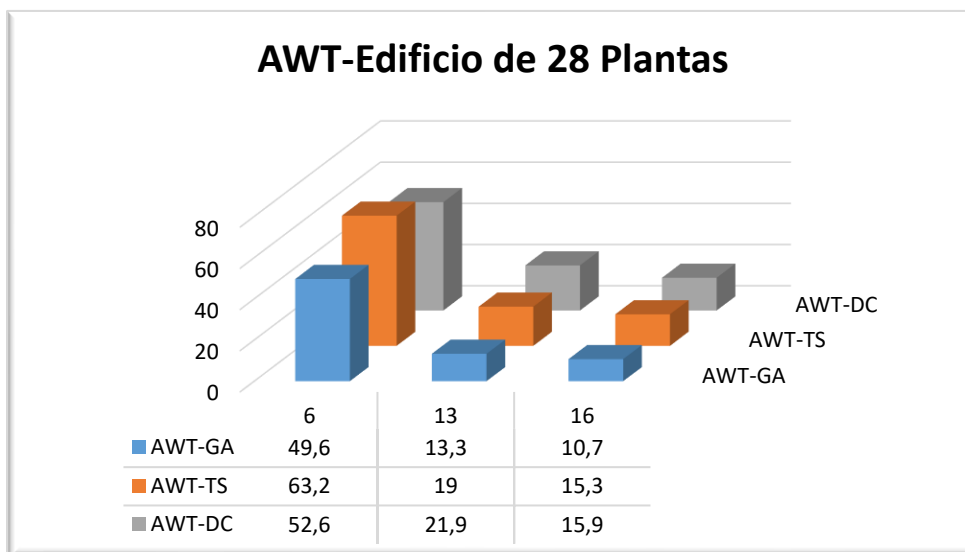


Figura 61: AWT (función de evaluación: distancia)- 28 Plantas. Lunchpeak

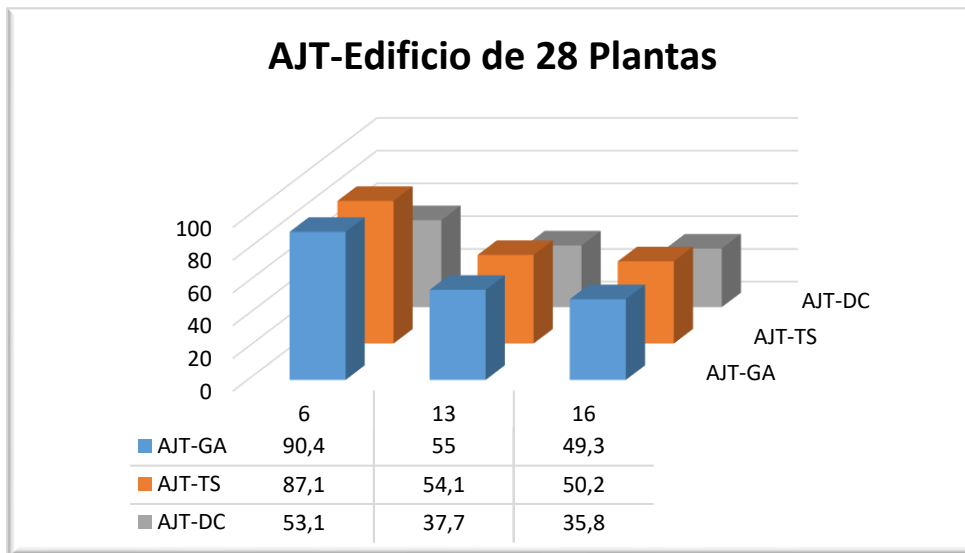


Figura 62: AJT (función de evaluación: distancia)- 28 Plantas. Lunchpeak

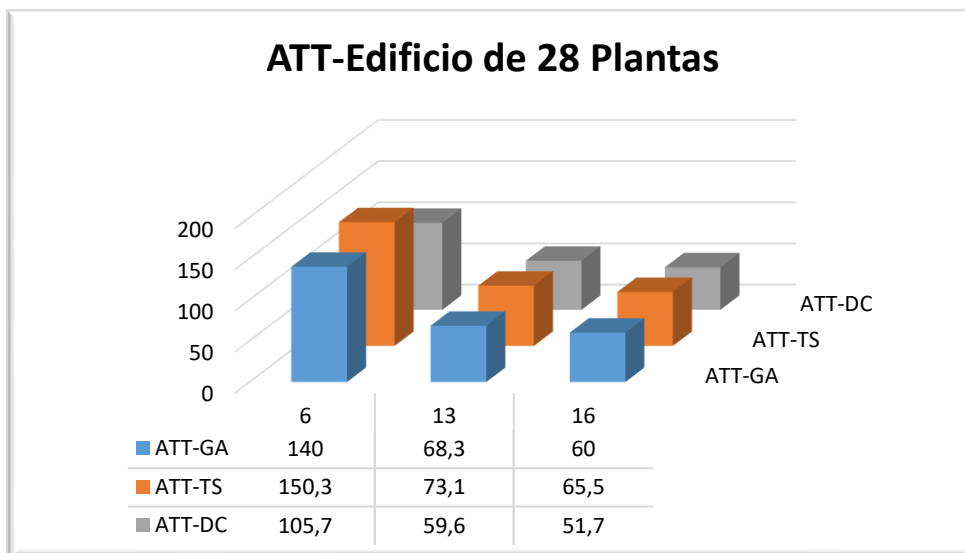


Figura 63: ATT (función de evaluación: distancia)- 28 Plantas. Lunchpeak

Para un edificio de 28 plantas se observa como el algoritmo genético muestra los mejores tiempos medios de espera, seguido por el algoritmo de búsqueda tabú. La Figura 62 muestra que el algoritmo DC-Elevate obtiene los mejores tiempos medios de tránsito. El tiempo total de viaje se minimiza con el algoritmo implementado por *Elevate*.

Edificio de 32 plantas

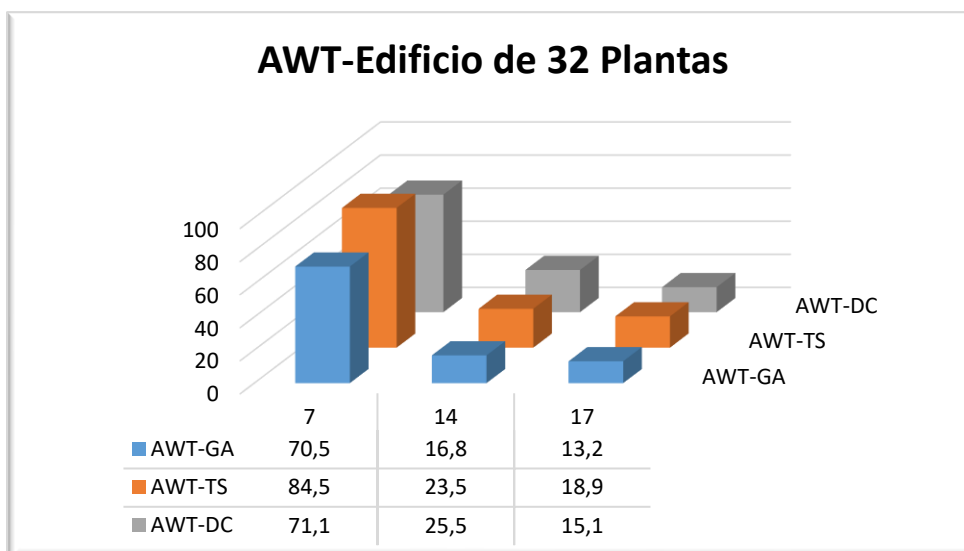


Figura 64: AWT (función de evaluación: distancia)- 32 Plantas. Lunchpeak

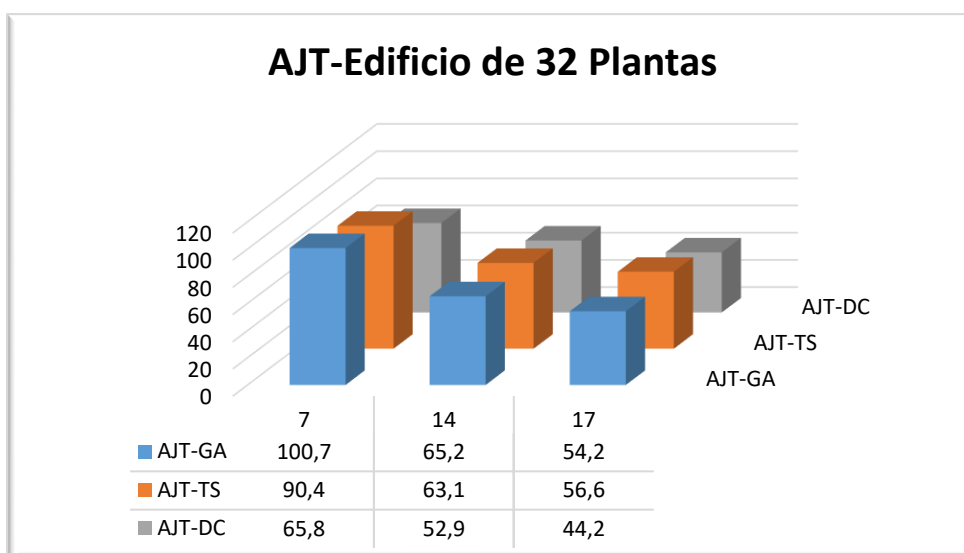


Figura 65: AJT (función de evaluación: distancia)- 32 Plantas. Lunchpeak

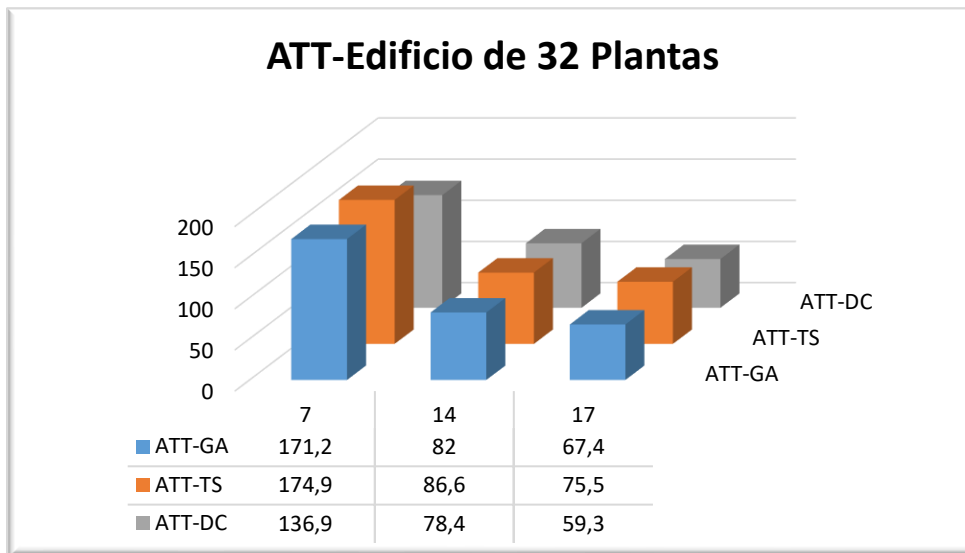


Figura 66: ATT (función de evaluación: distancia)- 32 Plantas. Lunchpeak

Para un edificio de 32 plantas se tiene que el AWT menor lo consigue el algoritmo genético mientras que la Figura 65 evidencia que el mejor AJT se tiene tras implementar DC-Elevate. Para los tiempos medios totales los tres algoritmos ofrecen resultados similares.

Edificio de 36 plantas

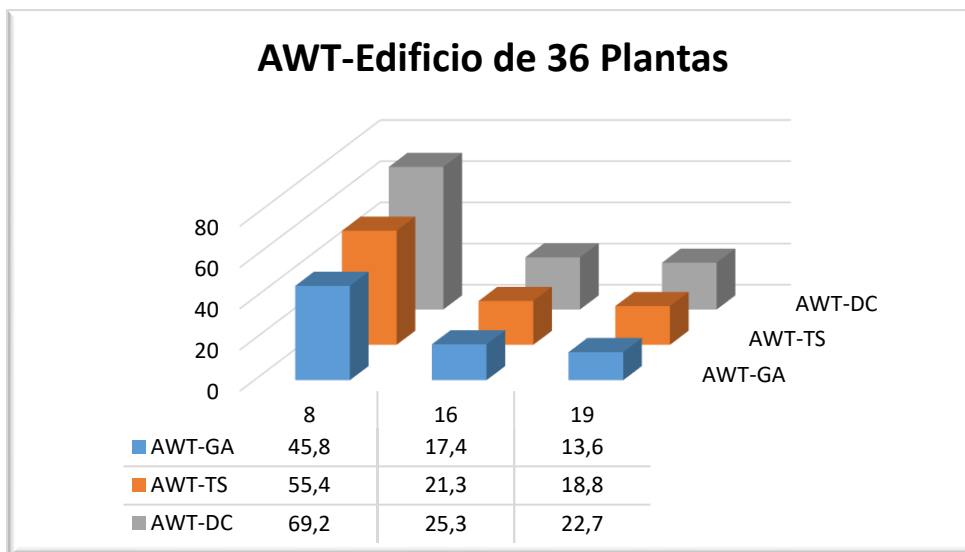


Figura 67: AWT (función de evaluación: distancia)- 36 Plantas. Lunchpeak

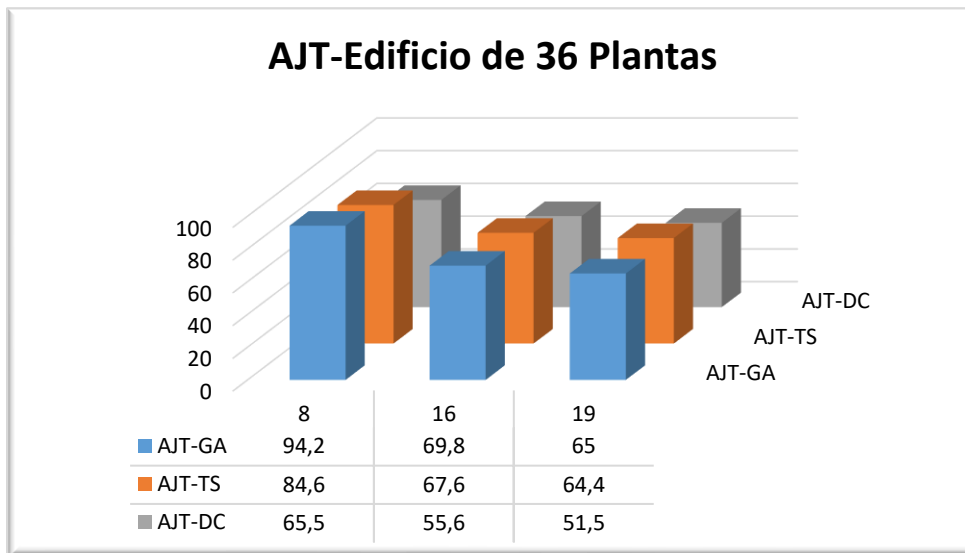


Figura 68: AJT (función de evaluación: distancia)- 36 Plantas. Lunchpeak

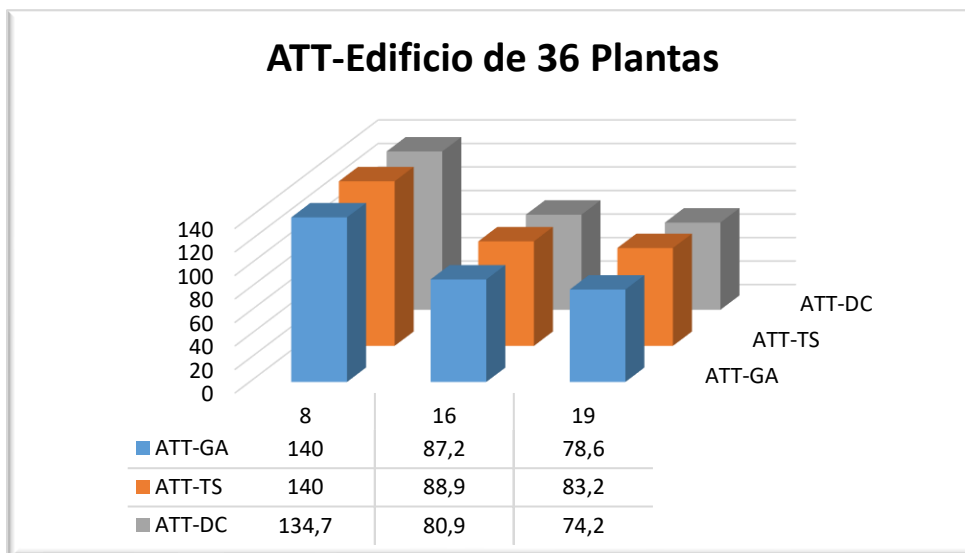


Figura 69: ATT (función de evaluación: distancia)- 36 Plantas. Lunchpeak

La Figura 67 evidencia que el algoritmo genético obtiene los mejores tiempos medios de espera de pasajeros para un edificio de 36 plantas. El tiempo medio de tránsito se minimiza mediante el empleo del algoritmo implementado por *Elevate*. La Figura 69 muestra que tanto el algoritmo genético como el algoritmo DC-Elevate ofrecen los mejores resultados de tiempos totales de viaje.

Edificio de 40 plantas

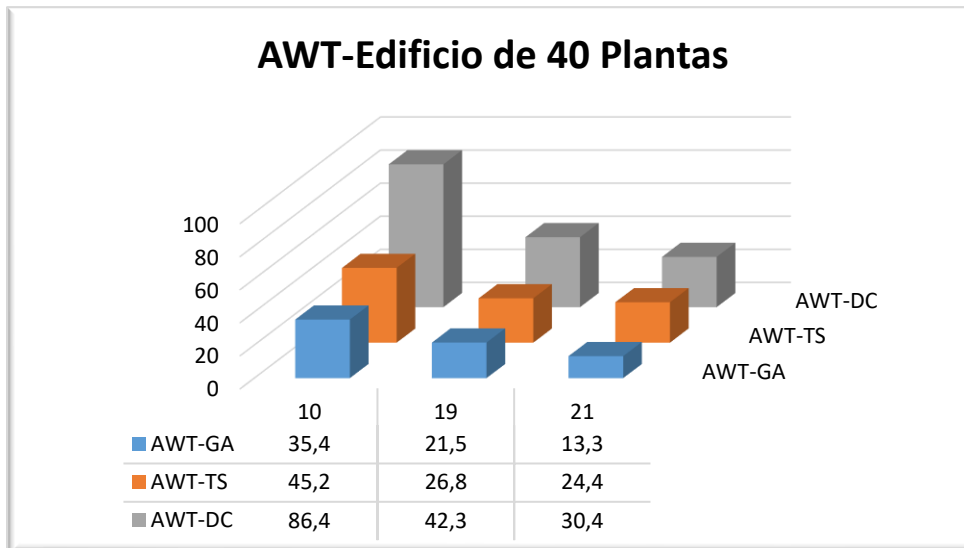


Figura 70: AWT (función de evaluación: distancia)- 40 Plantas. Lunchpeak

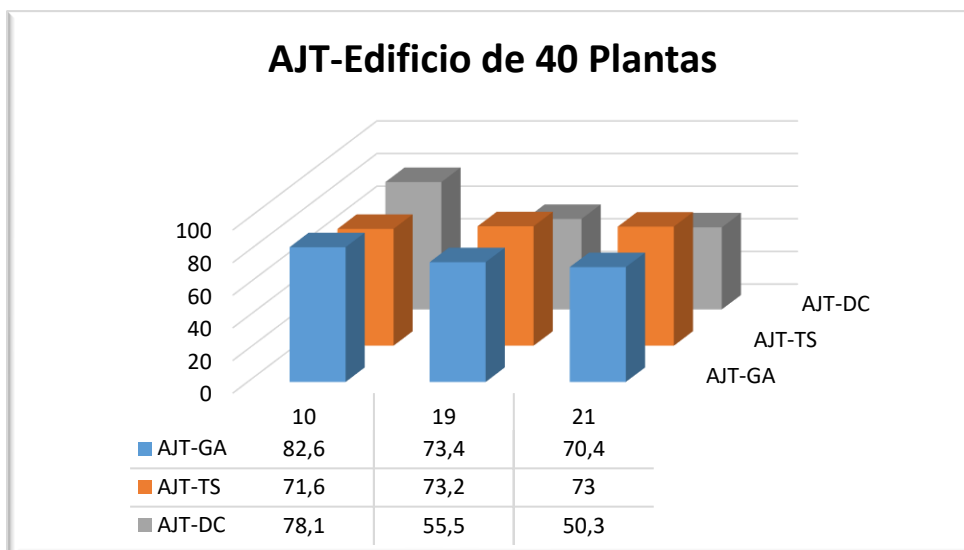


Figura 71: AJT (función de evaluación: distancia)- 40 Plantas. Lunchpeak

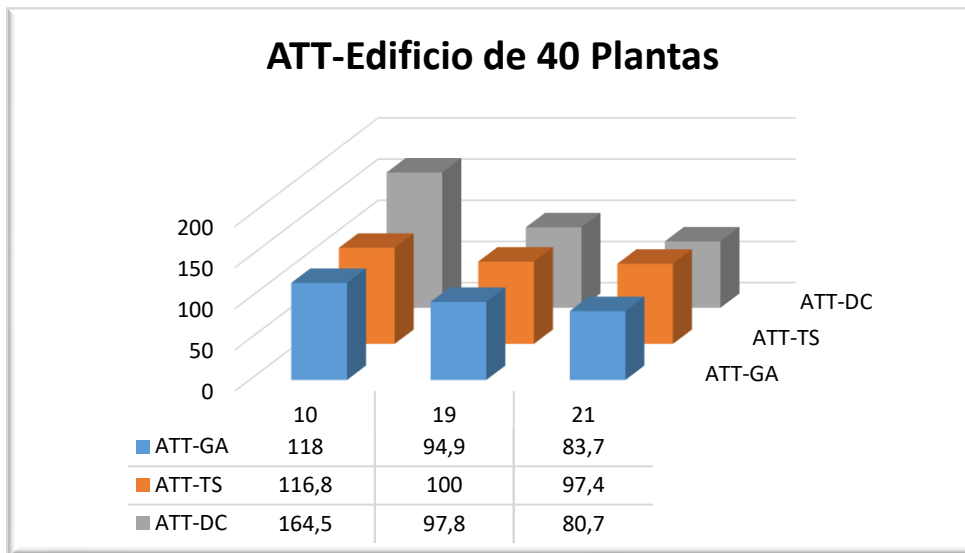


Figura 72: ATT (función de evaluación: distancia)- 40 Plantas. Lunchpeak

Para un edificio de 40 plantas, según la Figura 70, el tiempo medio de espera se minimiza mediante el algoritmo genético siguiéndolo el algoritmo de búsqueda tabú. Para el tiempo medio de tránsito se obtienen mejores valores mediante DC-Elevate aunque el resto de algoritmos no se diferencian demasiado. Para el ATT se consiguen valores parecidos para los tres algoritmos.

6.1.3. Tráfico Downpeak

Edificio de 12 plantas

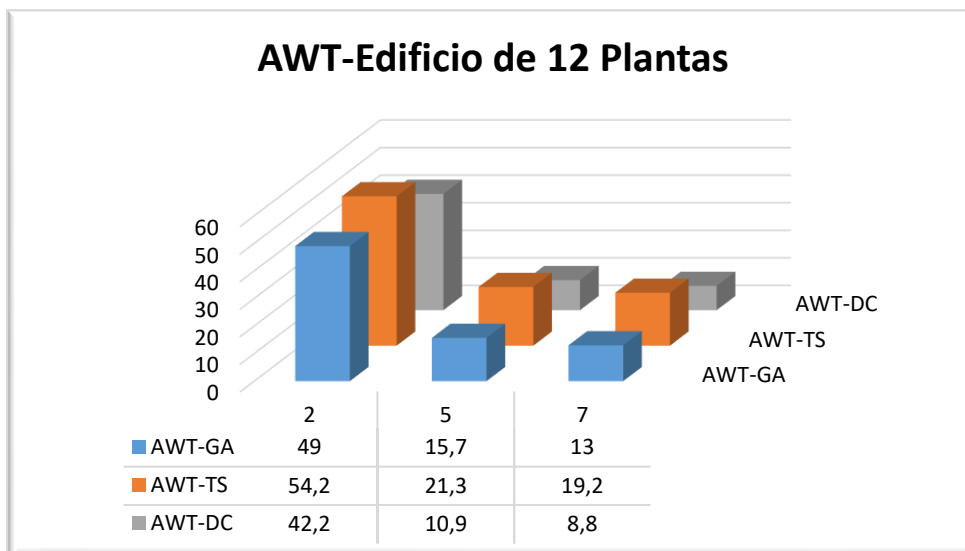


Figura 73: AWT (función de evaluación: distancia)- 12 Plantas. Downpeak

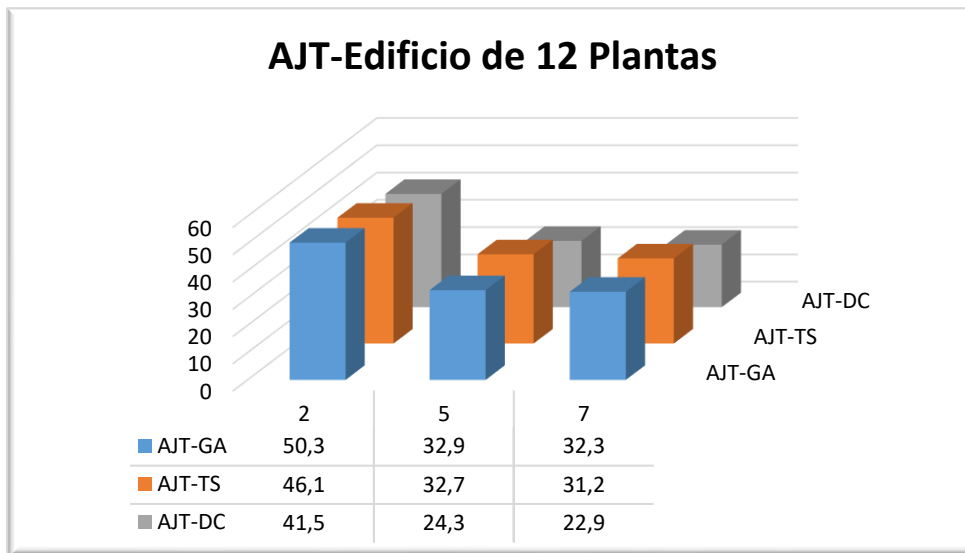


Figura 74: AJT (función de evaluación: distancia)- 12 Plantas. Downpeak

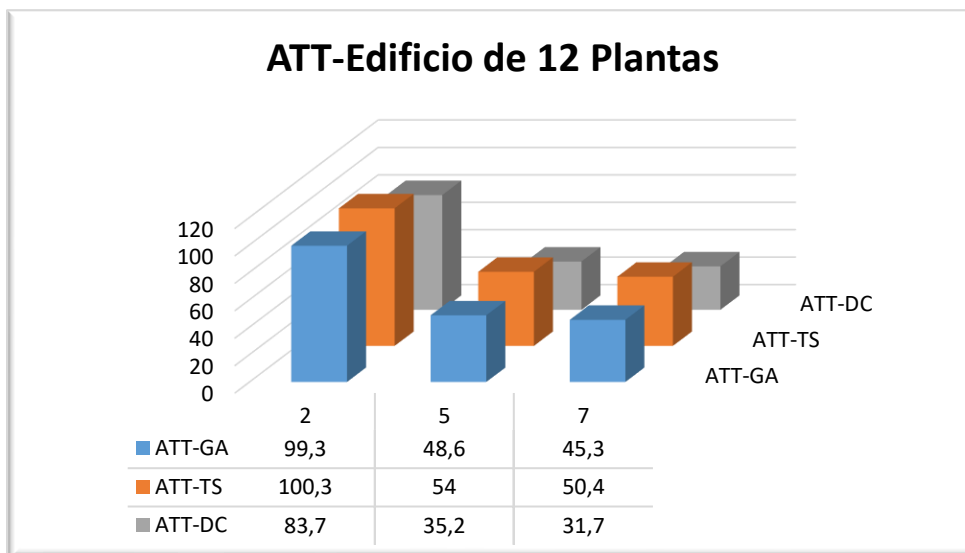


Figura 75: ATT (función de evaluación: distancia)- 12 Plantas. Downpeak

Para un edificio de 12 plantas se observa que el mejor tiempo medio de espera se obtiene mediante el algoritmo DC-Elevate. El tiempo medio de tránsito presenta un triple empate entre los algoritmos. El ATT evidencia en la Figura 75 que el algoritmo con mejores tiempos es el implementado por *Elevate*.

Edificio de 20 plantas

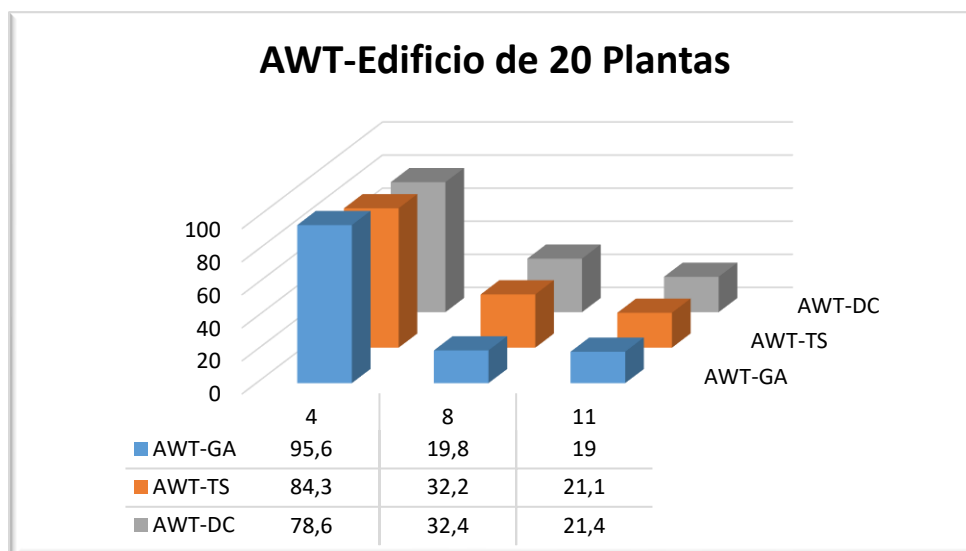


Figura 76: AWT (función de evaluación: distancia)- 20 Plantas. Downpeak

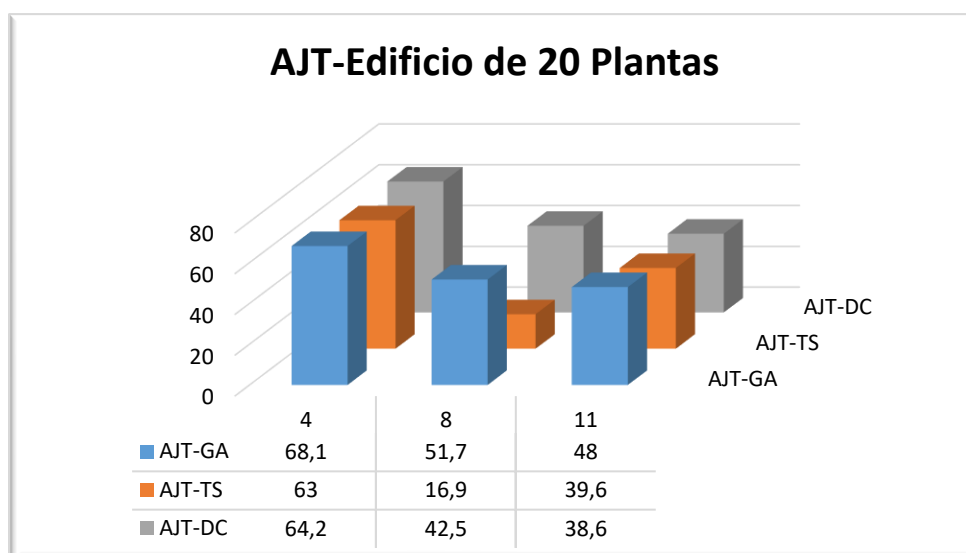


Figura 77: AJT (función de evaluación: distancia)- 20 Plantas. Downpeak

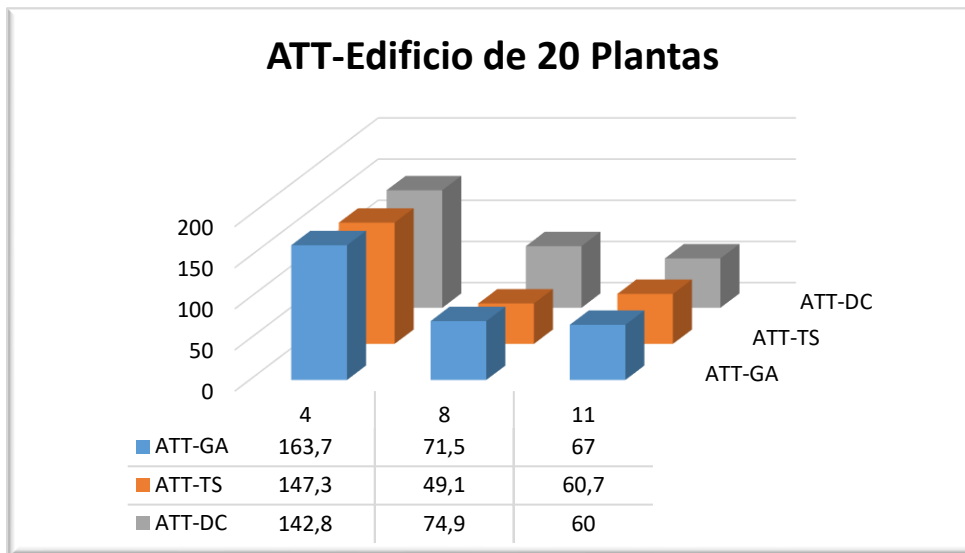


Figura 78: ATT (función de evaluación: distancia)- 20 Plantas. Downpeak

Para un edificio de 20 plantas se observa en la Figura 76 que el mejor tiempo medio de espera lo consigue el algoritmo genético seguido muy de cerca por los otros dos algoritmos. En cambio, el algoritmo de búsqueda tabú consigue los mejores resultados para el AJT. El tiempo total presenta un empate, aunque con algo de mejora sobre los otros dos por parte del algoritmo de búsqueda tabú.

Edificio de 24 plantas

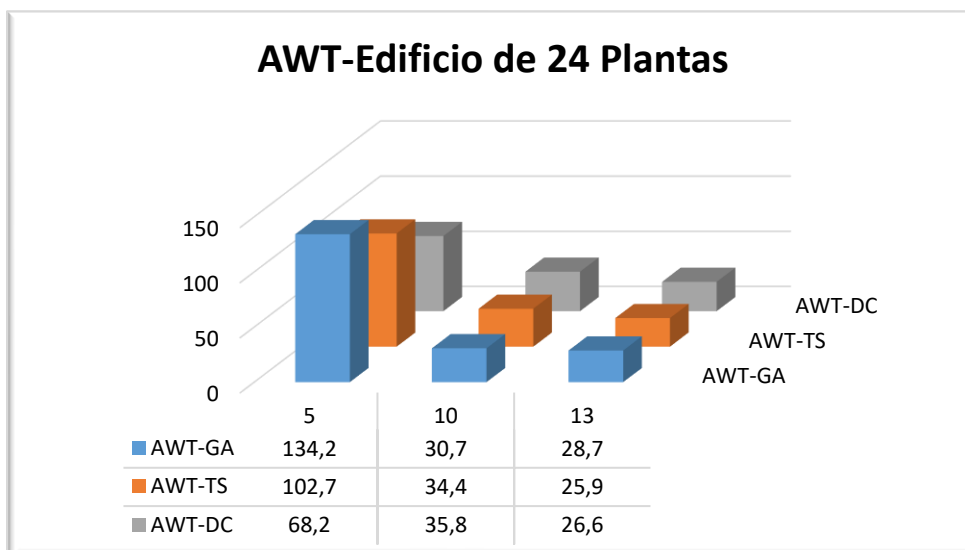


Figura 79: AWT (función de evaluación: distancia)- 24 Plantas. Downpeak

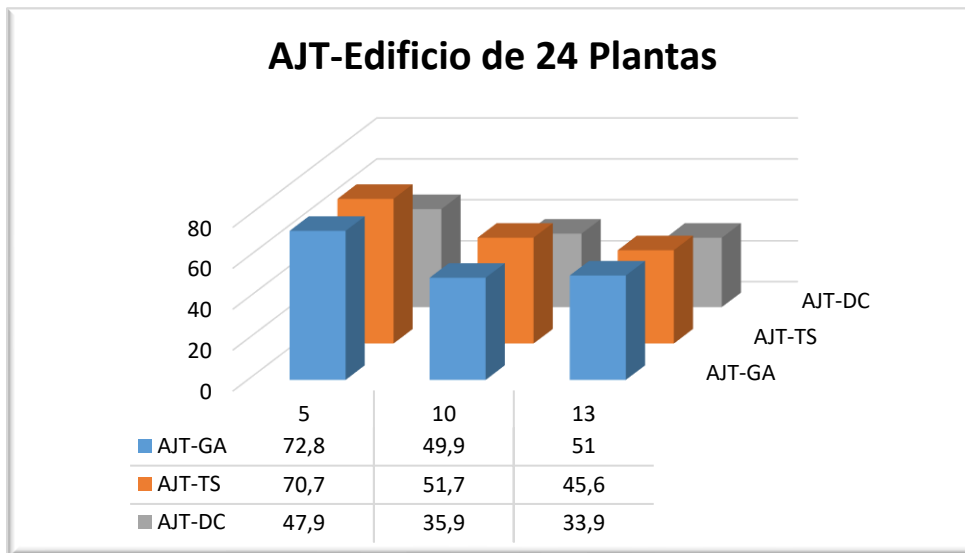


Figura 80: AJT (función de evaluación: distancia)- 24 Plantas. Downpeak

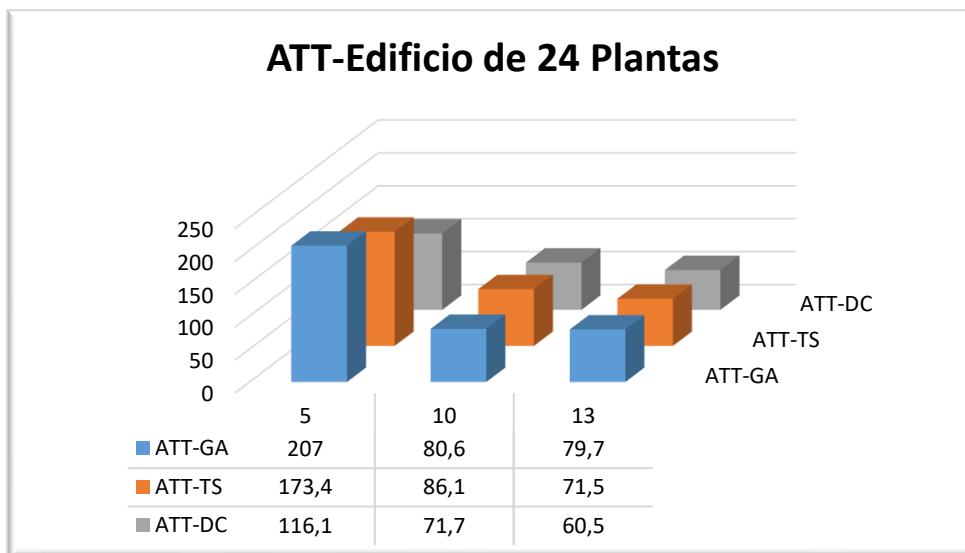


Figura 81: ATT (función de evaluación: distancia)- 24 Plantas. Downpeak

La Figura 79 arroja que los mejores tiempos medios de espera los presenta el algoritmo de búsqueda tabu para un edificio de 24 plantas. Sin embargo, el algoritmo DC-Elevate muestra los mejores tiempos medios de tránsito. Finalmente, se observa en la Figura 81 que existe un empate de tiempos totales, aunque con unos valores mejores para DC-Elevate.

Edificio de 28 plantas

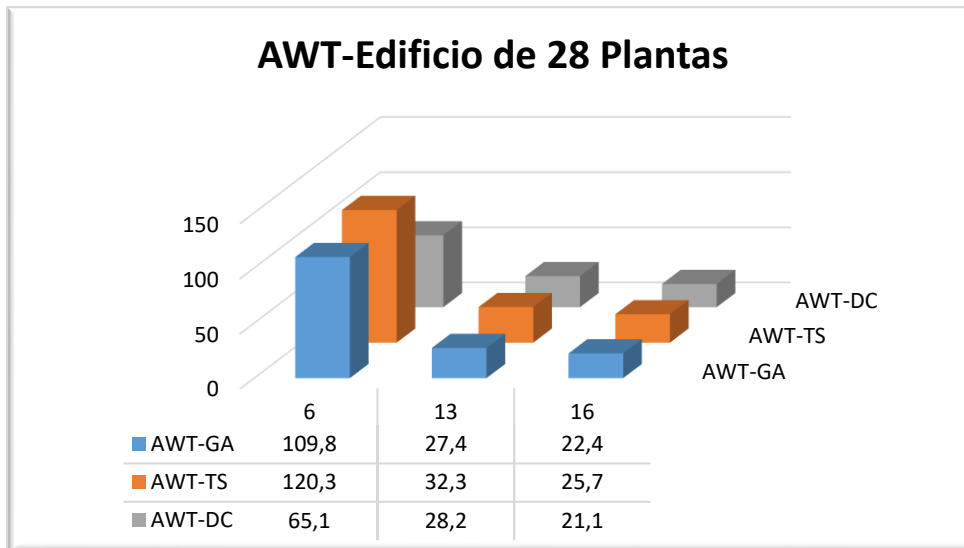


Figura 82: AWT (función de evaluación: distancia)- 24 Plantas. Downpeak

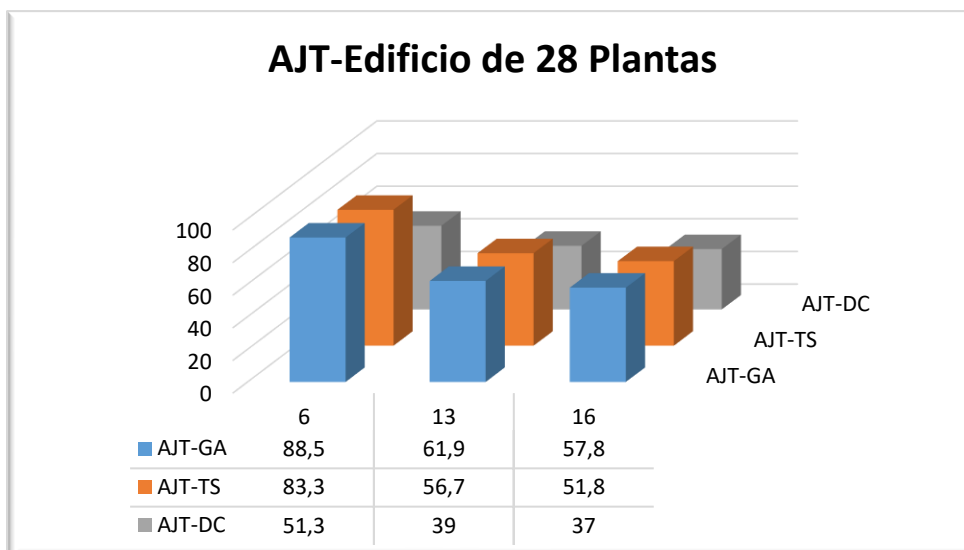


Figura 83: AJT (función de evaluación: distancia)- 28 Plantas. Downpeak

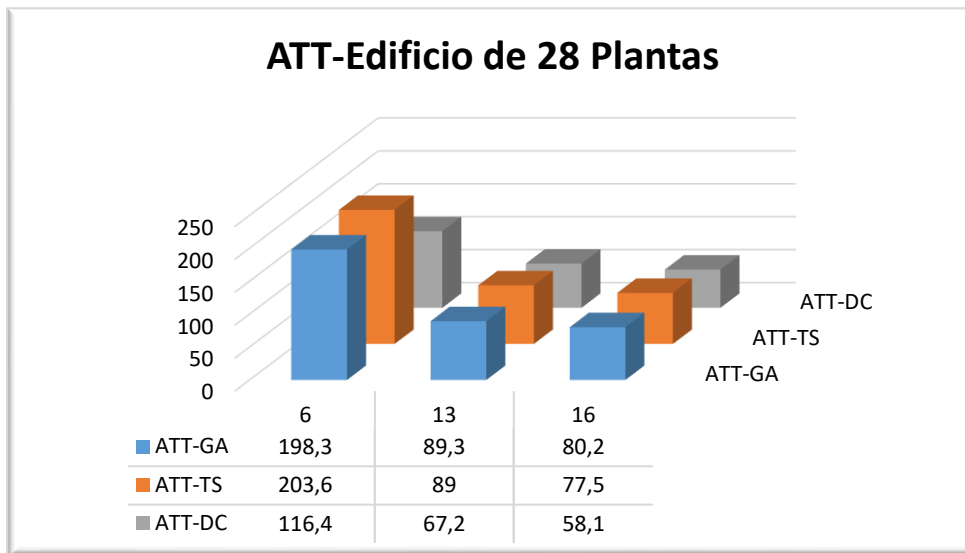


Figura 84: ATT (función de evaluación: distancia)- 28 Plantas. Downpeak

Para un edificio de 28 plantas se observa en la Figura 82 que el algoritmo genético y el algoritmo DC-Elevate presentan los mejores AWT. El algoritmo implementado por *Elevate* arroja los mejores tiempos medios de tránsito por lo que finalmente también consigue los mejores tiempos totales.

Edificio de 32 plantas

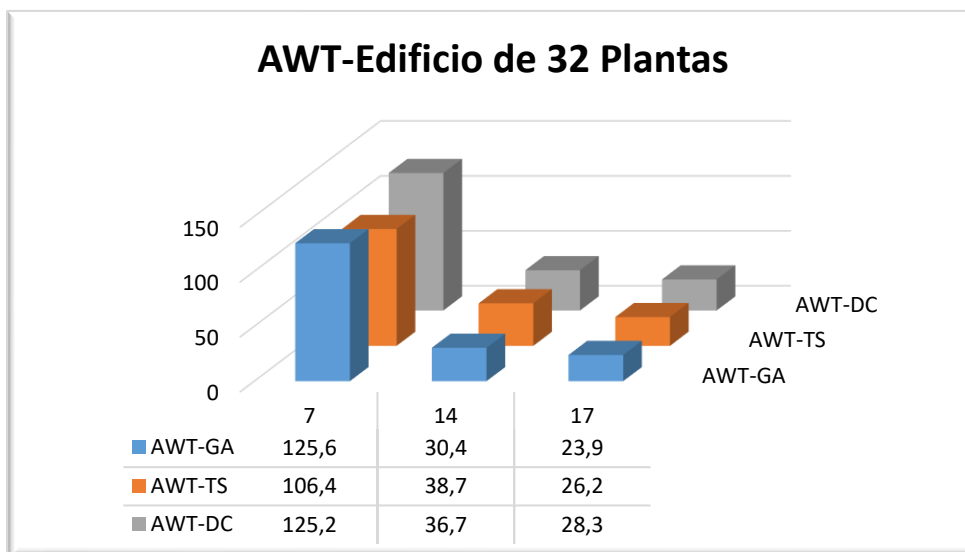


Figura 85: AWT (función de evaluación: distancia)- 32 Plantas. Downpeak

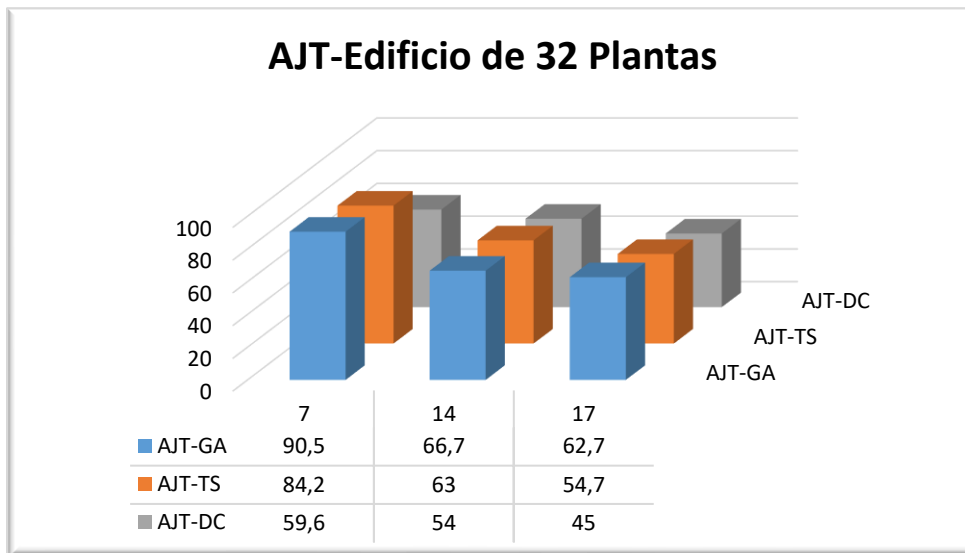


Figura 86: AJT (función de evaluación: distancia)- 32 Plantas. Downpeak

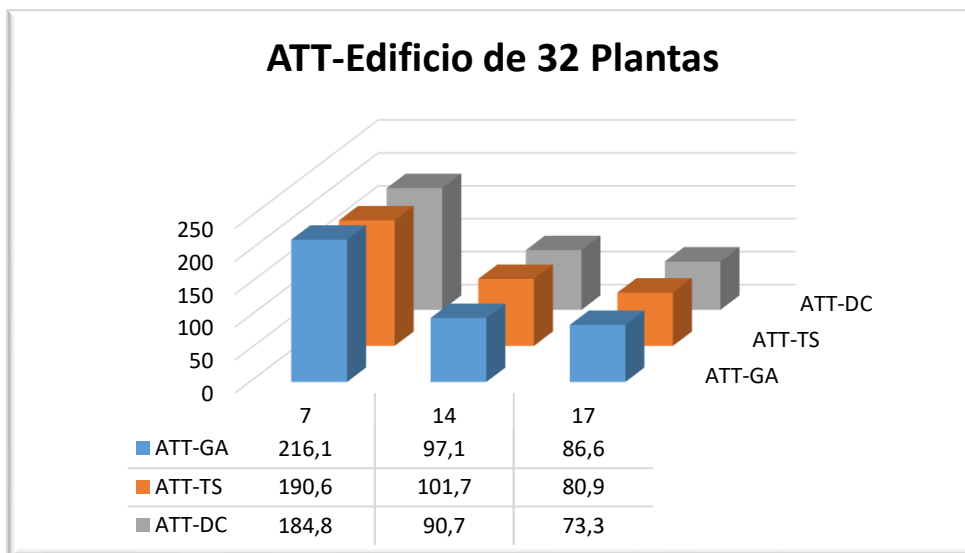


Figura 87: ATT (función de evaluación: distancia)- 32 Plantas. Downpeak

El algoritmo genético muestra los mejores tiempos medios de espera para un edificio de 32 plantas mientras que es el algoritmo DC-Elevate el que presenta mejores tiempos medios de tránsito según la Figura 86. Para el tiempo medio total se observa un empate entre los tres algoritmos en la Figura 87.

Edificio de 36 plantas

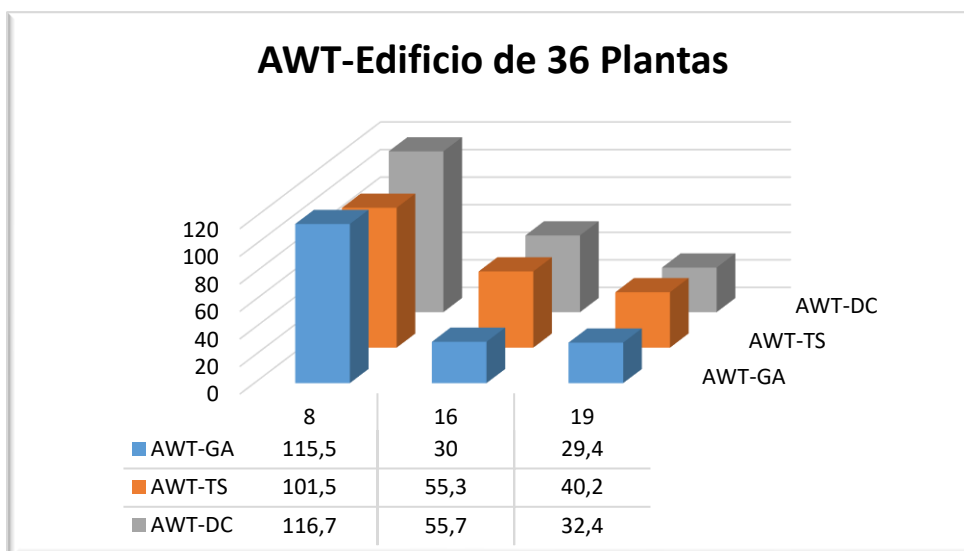


Figura 88: AWT (función de evaluación: distancia)- 36 Plantas. Downpeak

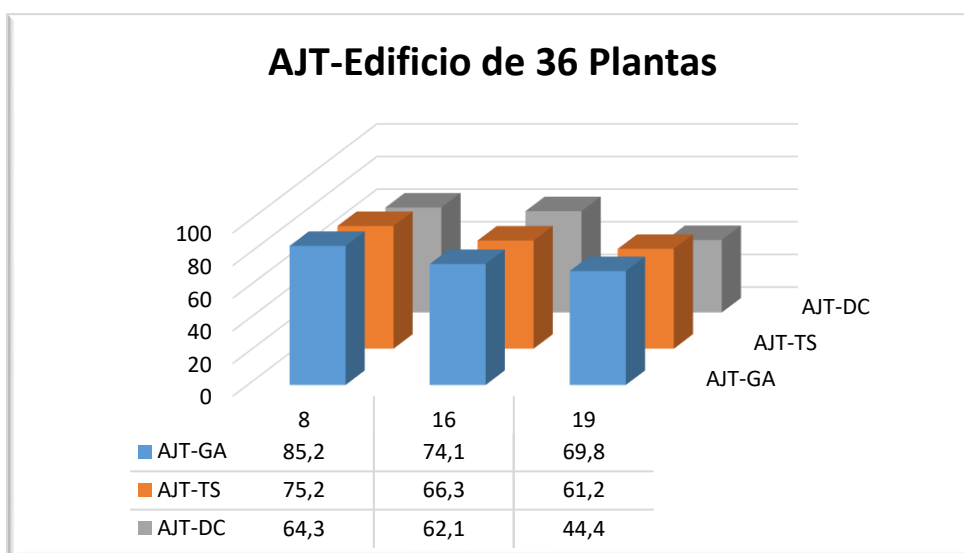


Figura 89: AJT (función de evaluación: distancia)- 36 Plantas. Downpeak

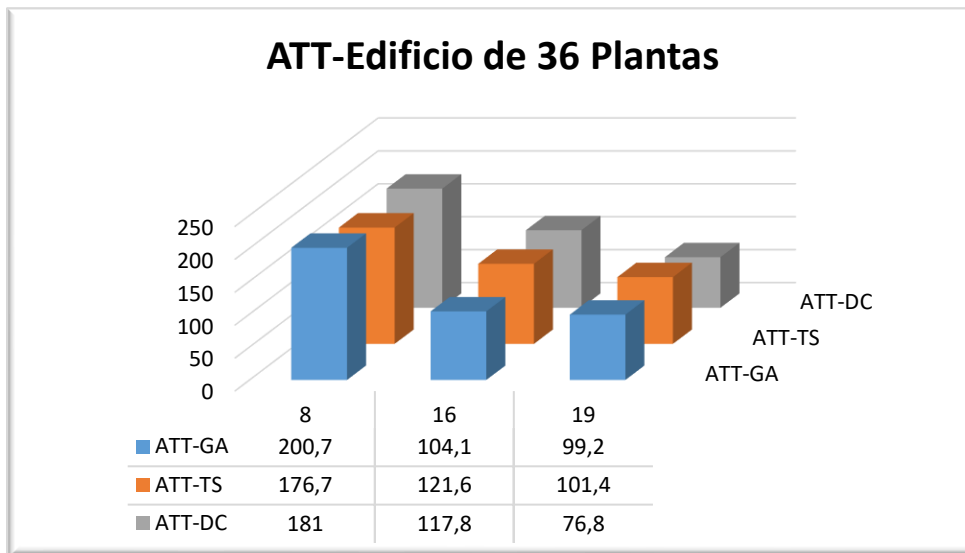


Figura 90: ATT (función de evaluación: distancia)- 36 Plantas. Downpeak

El edificio de 36 plantas presenta en la Figura 88 un tiempo medio de espera con mayor bondad para el algoritmo genético. Los mejores tiempos medios de tránsito los obtiene el algoritmo DC-Elevate. El ATT obtiene los mejores valores para el algoritmo DC-Elevate y el algoritmo genético.

Edificio de 40 plantas

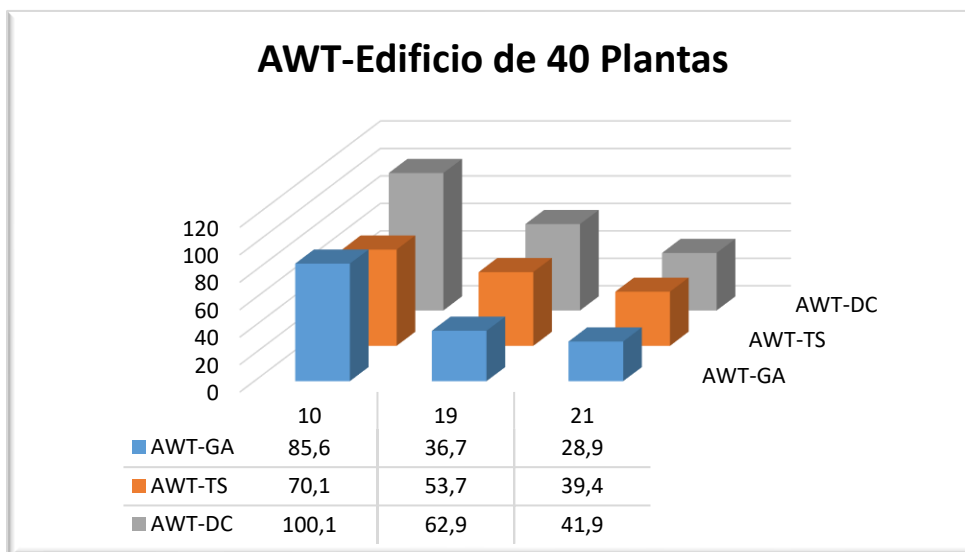


Figura 91: AWT (función de evaluación: distancia)- 40 Plantas. Downpeak

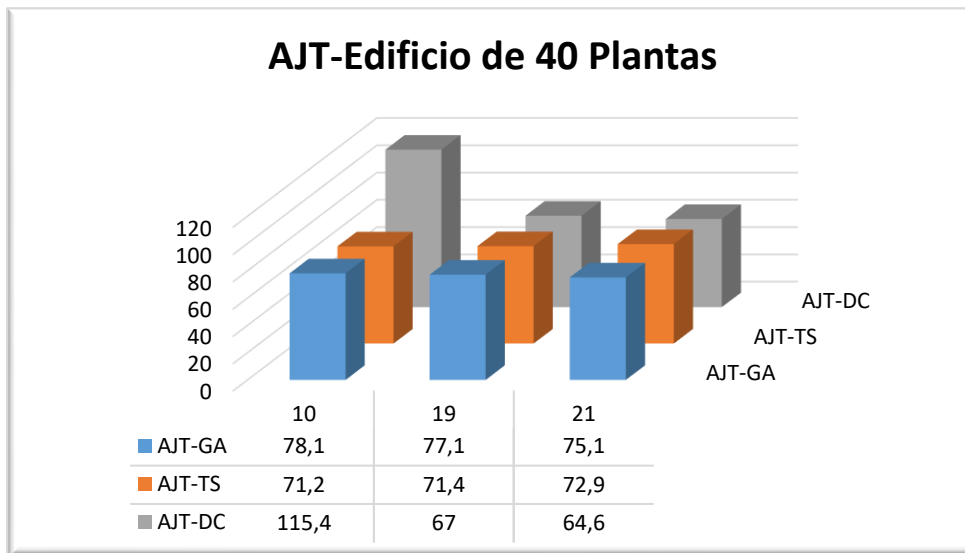


Figura 92: AJT (función de evaluación: distancia)- 40 Plantas. Downpeak

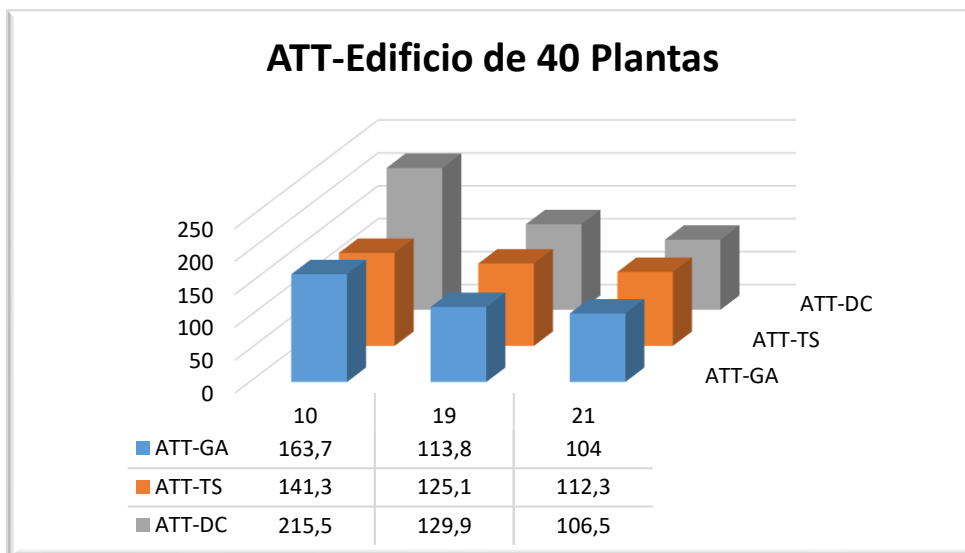


Figura 93: ATT (función de evaluación: distancia)- 40 Plantas. Downpeak

Para un edificio de 40 plantas se muestra en la Figura 91 que el tiempo medio de espera que ofrece el algoritmo genético es el más bajo. El mejor tiempo medio de tránsito se obtiene entre el algoritmo de búsqueda tabú y el implementado por *Elevate*. La Figura 93 muestra como el algoritmo genético y el algoritmo DC-Elevate presentan los mejores tiempos totales medios.

6.2. Función de evaluación basada en tiempo

Para cada tipo de tráfico se analiza el comportamiento del tráfico vertical en función de la cantidad de plantas y ascensores que dispone el edificio, así como del método de resolución.

6.2.1. Tráfico Interfloor

Edificio de 12 plantas

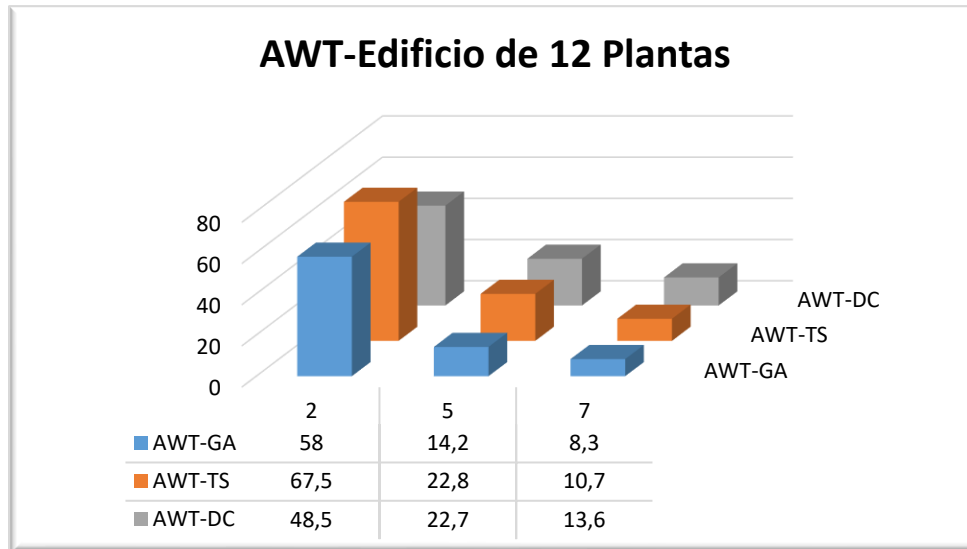


Figura 94: AWT (función de evaluación: tiempo)- 12 Plantas. Interfloor

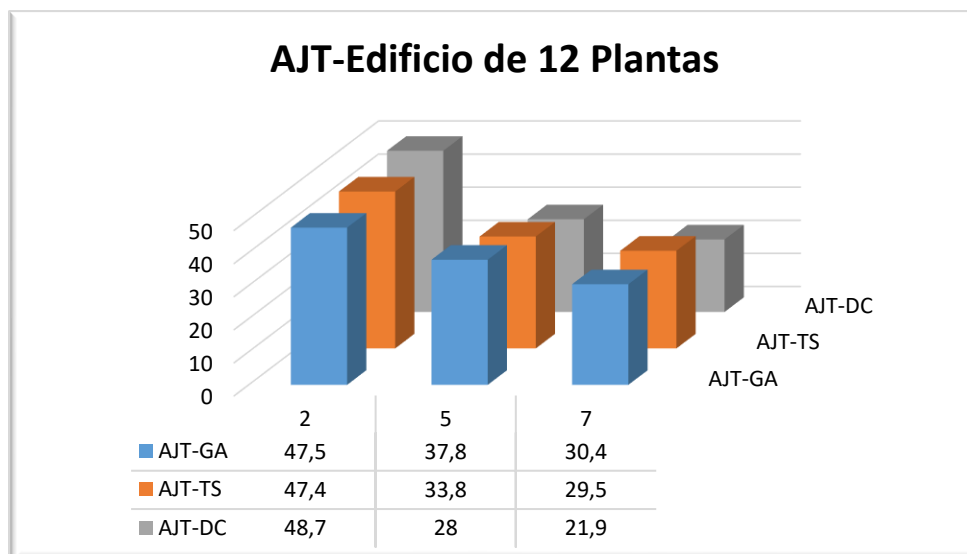


Figura 95: AJT (función de evaluación: tiempo)- 12 Plantas. Interfloor

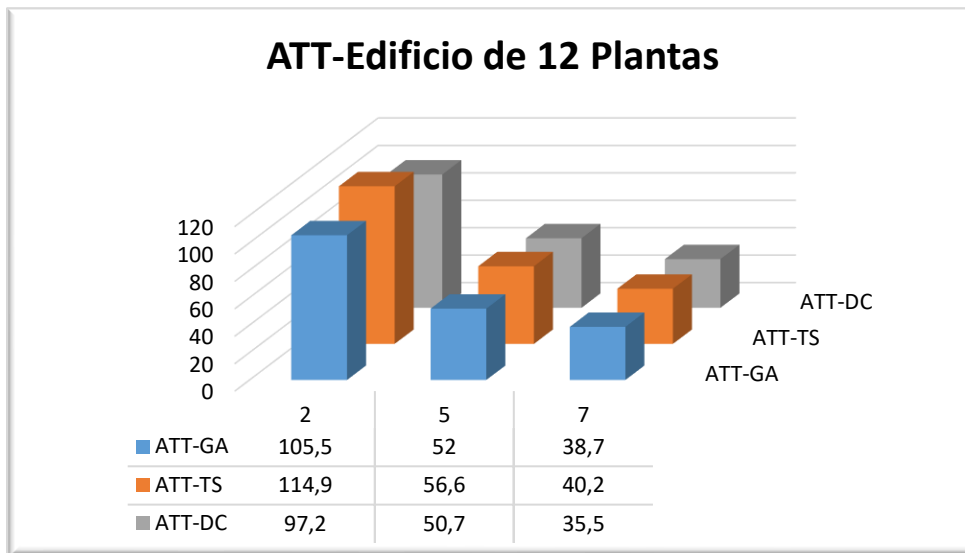


Figura 96: ATT (función de evaluación: tiempo)- 12 Plantas. Interfloor

Para un edificio de 12 plantas se observa que el algoritmo genético presenta los mejores tiempos medios de espera de pasajeros mientras que el tiempo medio de viaje lo obtiene con mayor bondad el algoritmo implementado por *Elevate* seguido del algoritmo de búsqueda tabú. En la Figura 96 se aprecia que tanto el algoritmo genético como DC-Elevate son los que presentan mejores tiempos medios totales.

Edificio de 20 plantas

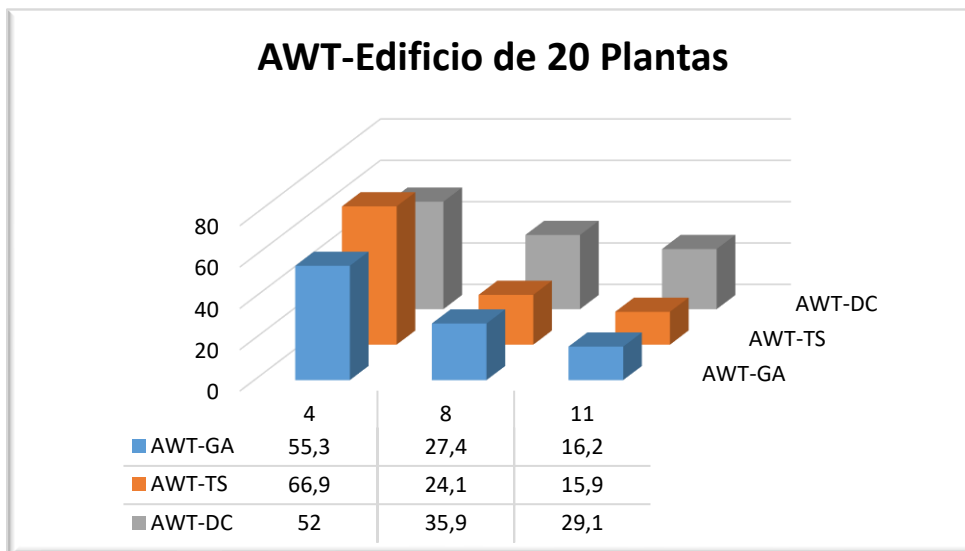


Figura 97: AWT (función de evaluación: tiempo)- 20 Plantas. Interfloor

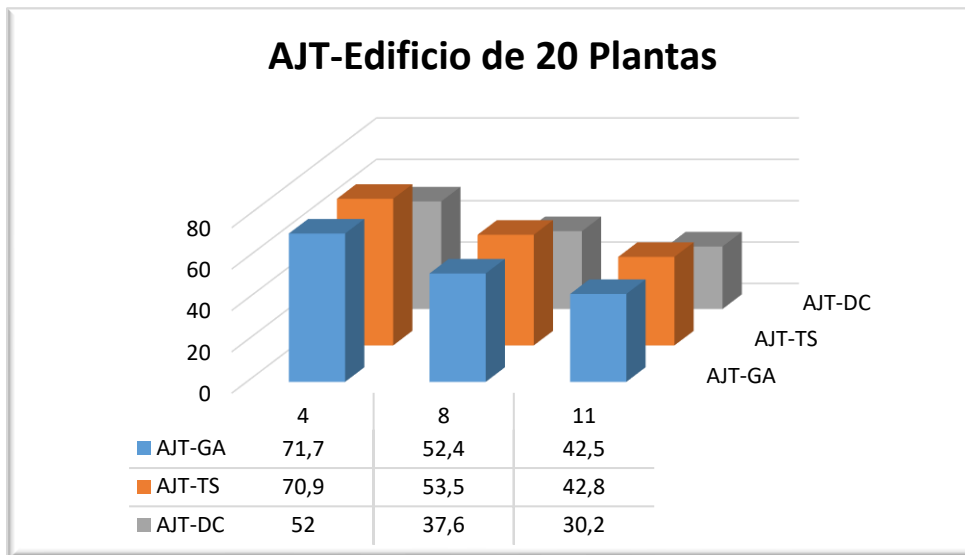


Figura 98: AJT (función de evaluación: tiempo)- 20 Plantas. Interfloor

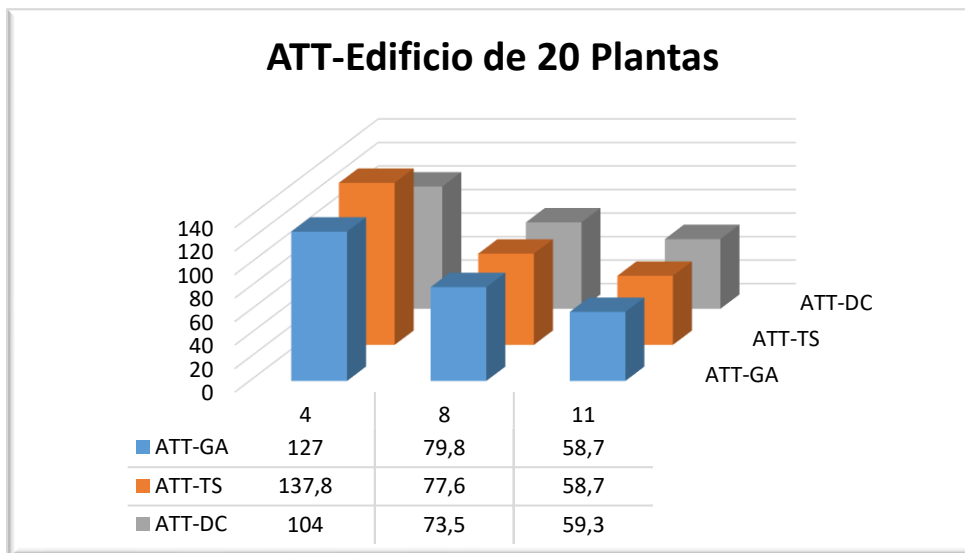


Figura 99: ATT (función de evaluación: tiempo)- 20 Plantas. Interfloor

En un edificio de 20 plantas se observa en la Figura 97 que tanto el algoritmo genético como el algoritmo de búsqueda tabú obtienen los mejores tiempos medios. En cambio, DC-Elevate obtiene los mejores tiempos medios de tránsito según la Figura 98. Para el tiempo medio total, los mejores valores se obtienen a modo de empate entre los tres algoritmos.

Edificio de 24 plantas

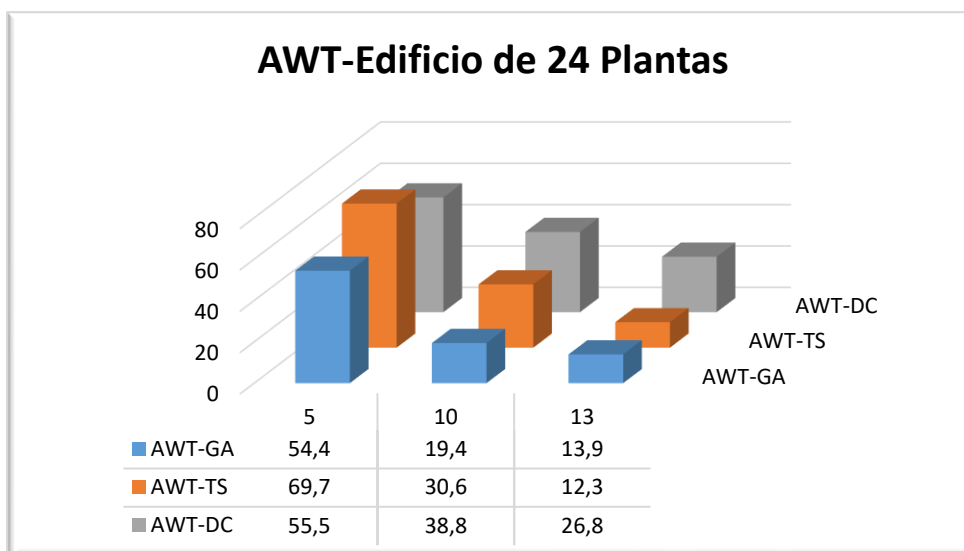


Figura 100: AWT (función de evaluación: tiempo)- 24 Plantas. Interfloor

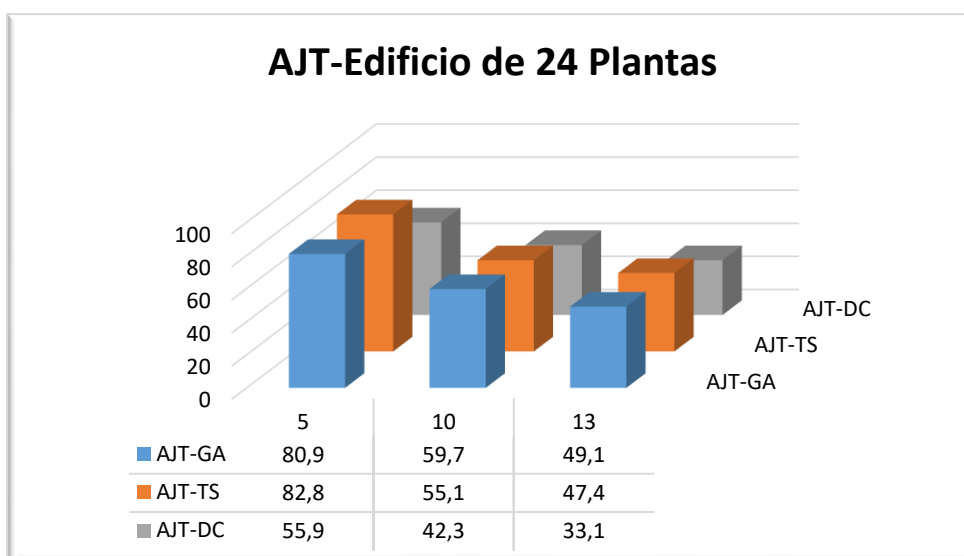


Figura 101: AJT (función de evaluación: tiempo)- 24 Plantas. Interfloor

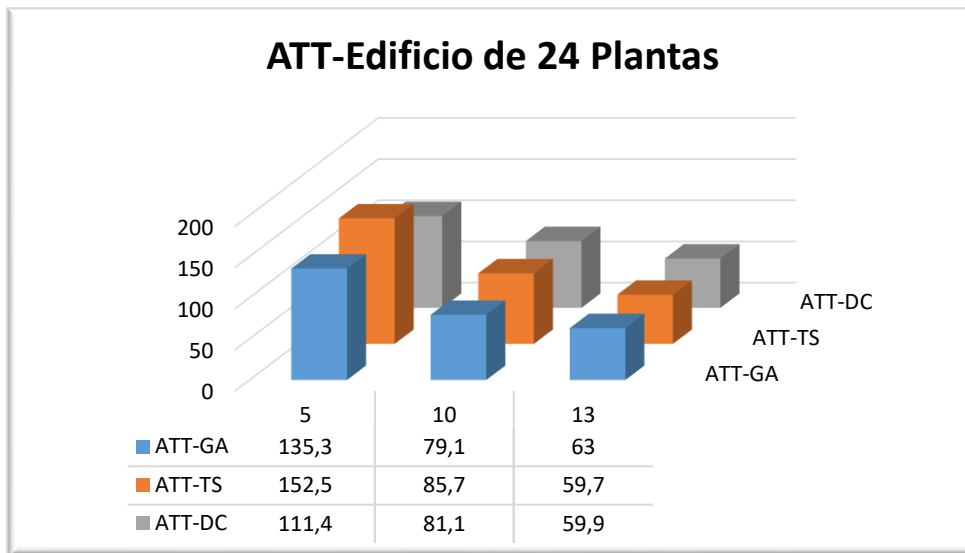


Figura 102: ATT (función de evaluación: tiempo)- 24 Plantas. Interfloor

La Figura 100 presenta a la búsqueda tabú y al algoritmo genético como los algoritmos que obtienen los mejores tiempos medios de espera. Sin embargo, es el algoritmo implementado por *Elevate* el que obtiene los mejores tiempos medios de tránsito. Se observa en la Figura 102 que el algoritmo genético obtiene el mejor tiempo medio total.

Edificio de 28 plantas

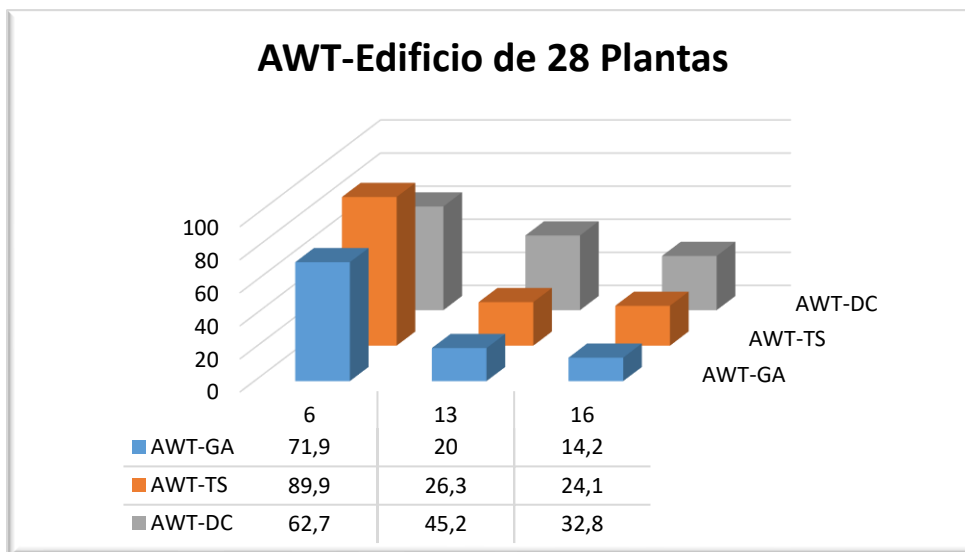


Figura 103: AWT (función de evaluación: tiempo)- 28 Plantas. Interfloor

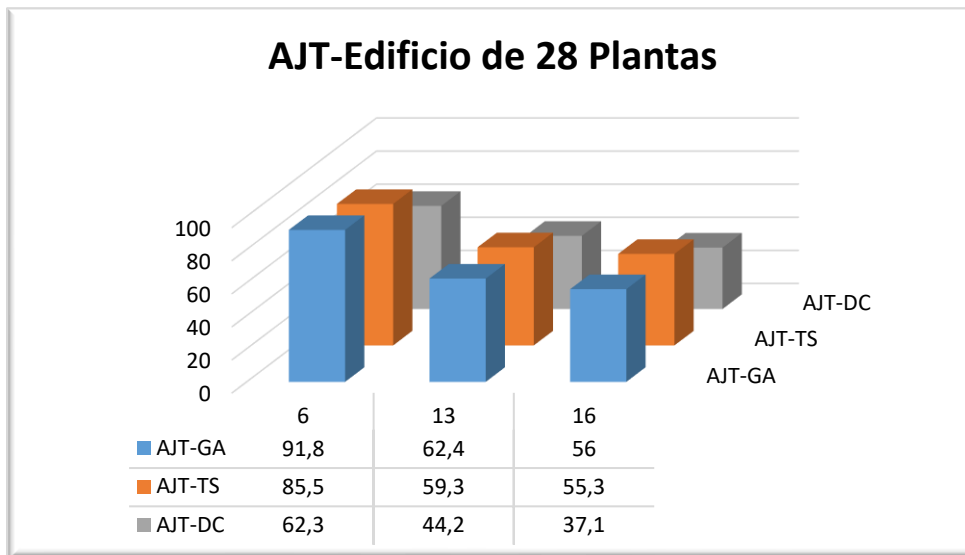


Figura 104: AJT (función de evaluación: tiempo)- 28 Plantas. Interfloor

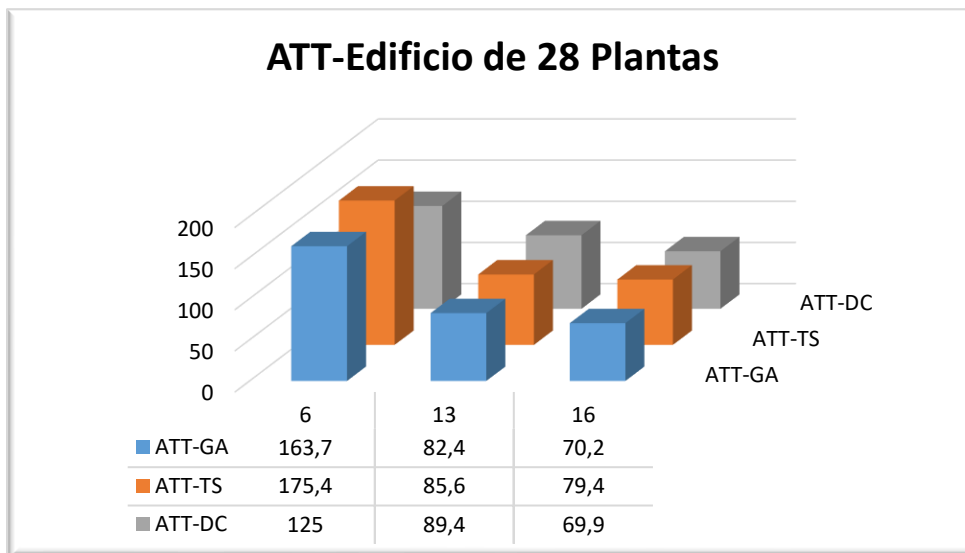


Figura 105: ATT (función de evaluación: tiempo)- 28 Plantas. Interfloor

Para un edificio de 28 plantas se muestra en la Figura 103 como el algoritmo genético obtiene los mejores tiempos medios de espera. Para el tiempo medio de tránsito se obtiene el mejor valor mediante DC-Elevate. La Figura 105 refleja como el algoritmo genético y el algoritmo de búsqueda tabú obtienen los mejores tiempos medios totales.

Edificio de 32 plantas

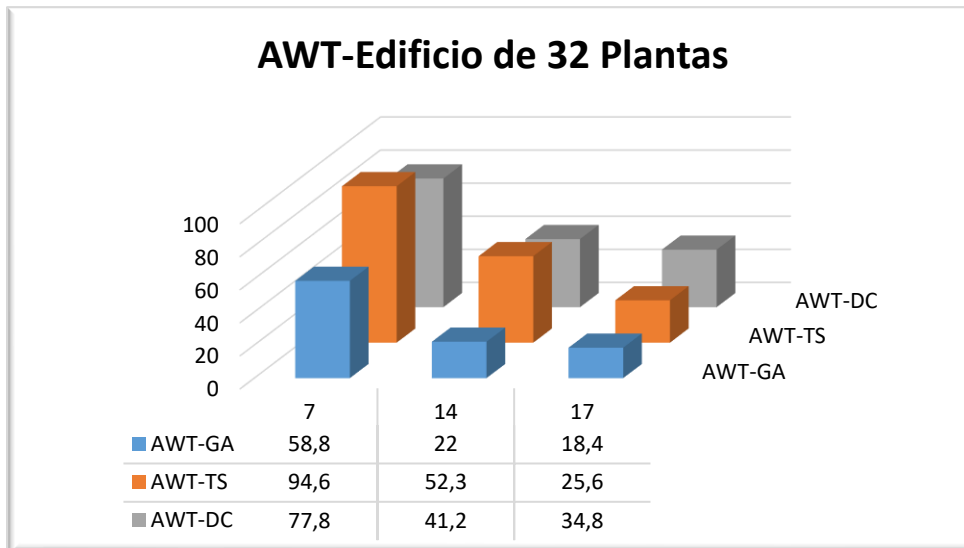


Figura 106: AWT (función de evaluación: tiempo)- 32 Plantas. Interfloor

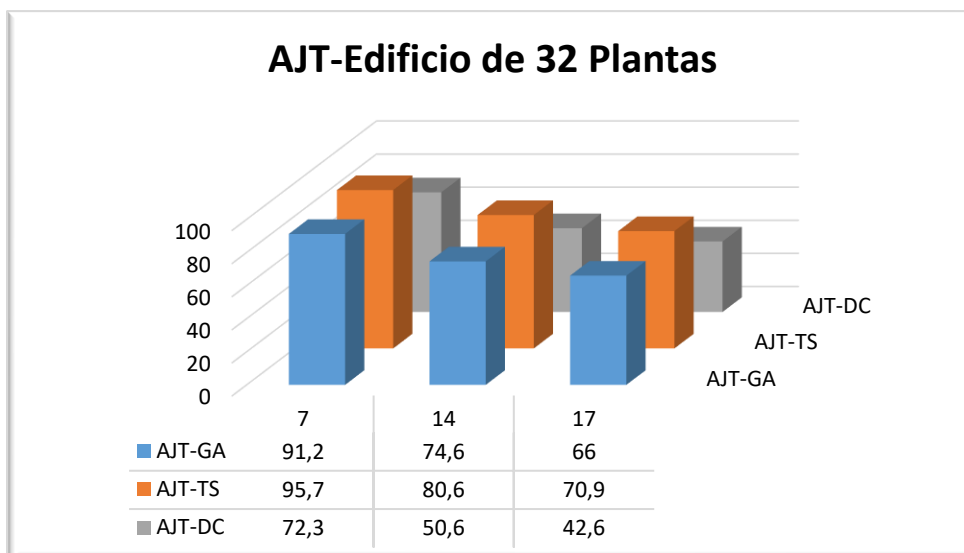


Figura 107: AJT (función de evaluación: tiempo)- 32 Plantas. Interfloor

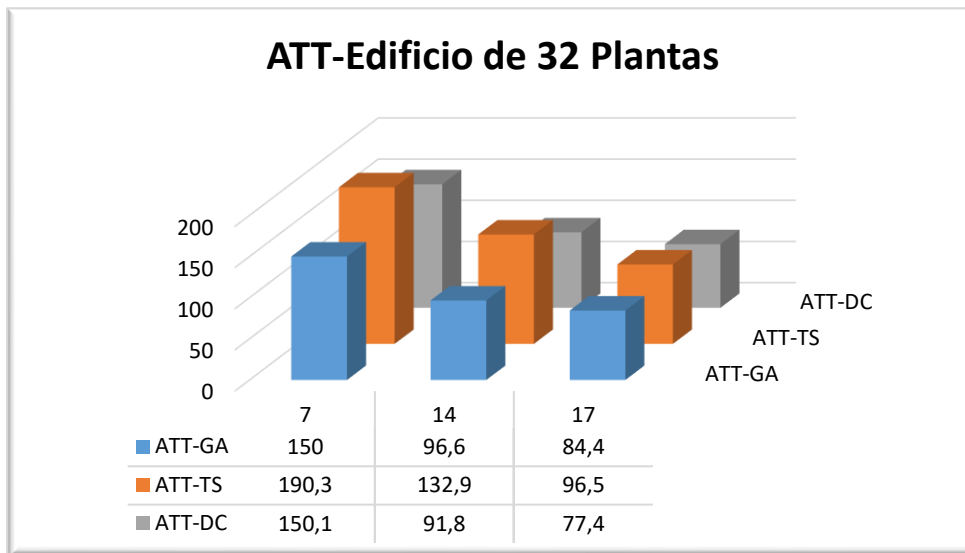


Figura 108: ATT (función de evaluación: tiempo)- 32 Plantas. Interfloor

La Figura 106 refleja que el algoritmo genético obtiene los mejores tiempos medios de espera. El tiempo medio de tránsito es minimizado por DC-Elevate seguido por el algoritmo genético. Para el tiempo medio total se tiene que tanto el algoritmo genético como el algoritmo implementado por *Elevate* muestran los mejores resultados según la Figura 108.

Edificio de 36 plantas

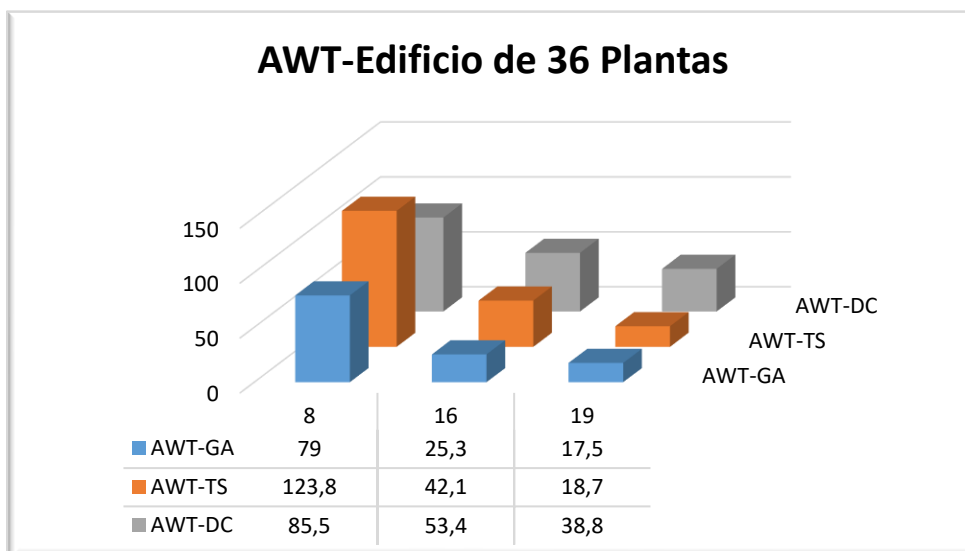


Figura 109: AWT (función de evaluación: tiempo)- 36 Plantas. Interfloor

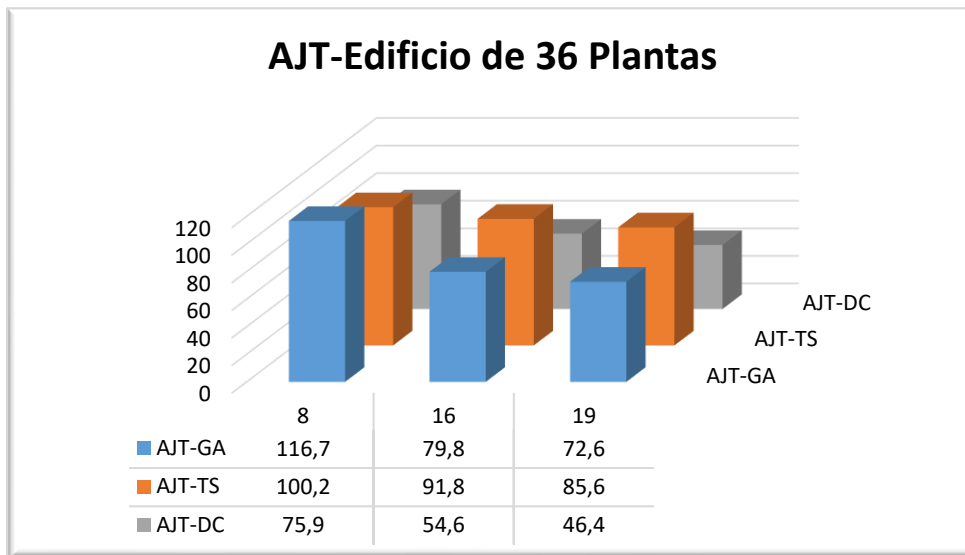


Figura 110: AJT (función de evaluación: tiempo)- 36 Plantas. Interfloor

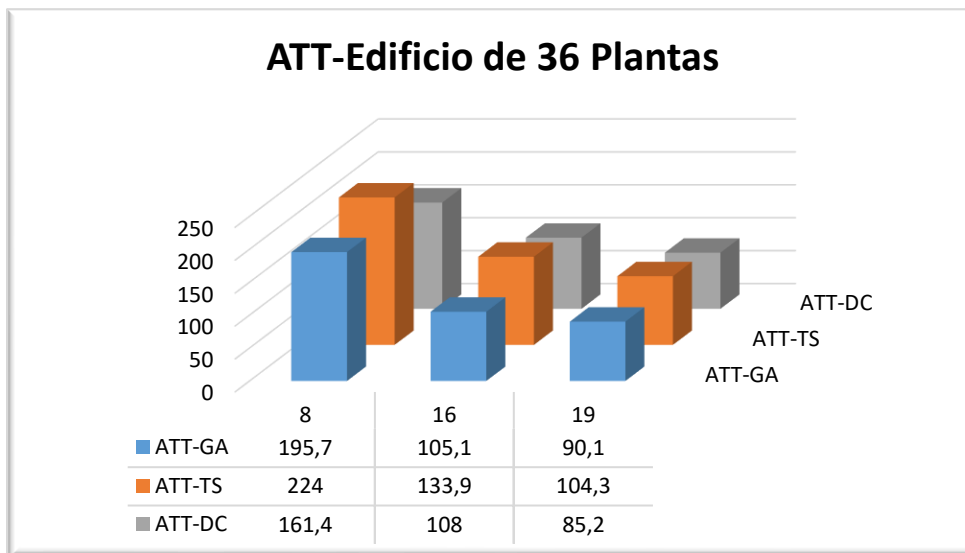


Figura 111: ATT (función de evaluación: tiempo)- 36 Plantas. Interfloor

Se muestra en la Figura 109 como el algoritmo genético consigue los mejores valores para el tiempo medio de espera junto con el algoritmo de búsqueda tabú. Es el algoritmo implementado por *Elevate* el que obtiene los mejores tiempos medios de tránsito mientras que el algoritmo genético obtiene el mejor tiempo medio total según la Figura 111.

Edificio de 40 plantas

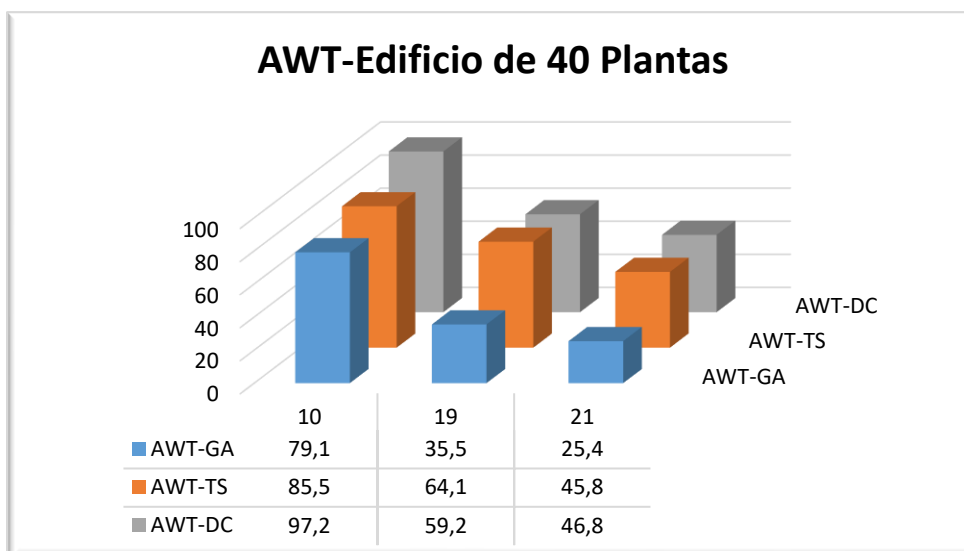


Figura 112: AWT (función de evaluación: tiempo)- 40 Plantas. Interfloor

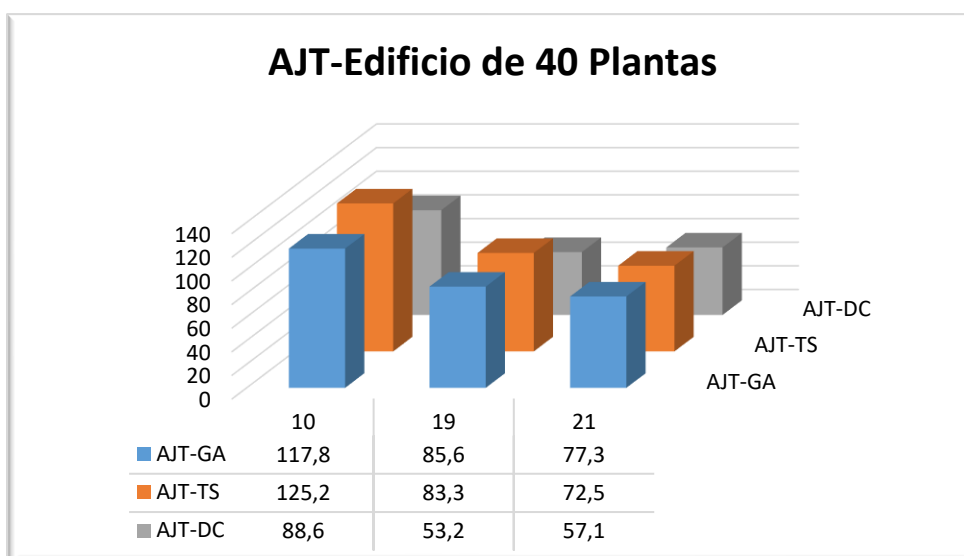


Figura 113: AJT (función de evaluación: tiempo)- 40 Plantas. Interfloor

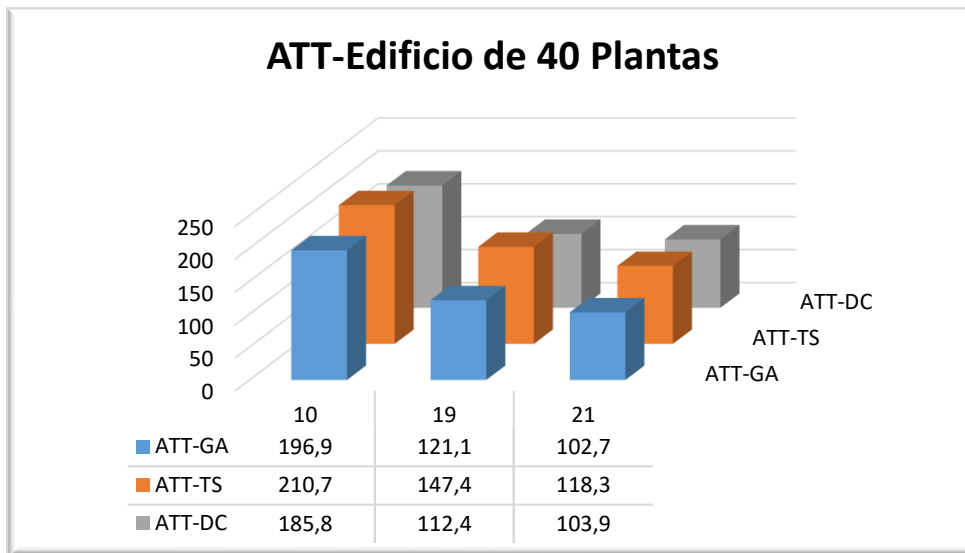


Figura 114: ATT (función de evaluación: tiempo)- 40 Plantas. Interfloor

Para un edificio de 40 plantas se muestra en la Figura 112 que el algoritmo genético obtiene los mejores AWT frente al algoritmo implementado por *Elevate* que consigue los mejores AJT. Sin embargo, los mejores ATT son obtenidos mediante el algoritmo genético como se muestra en la Figura 114.

6.2.2. Tráfico Lunchpeak

Edificio de 12 plantas

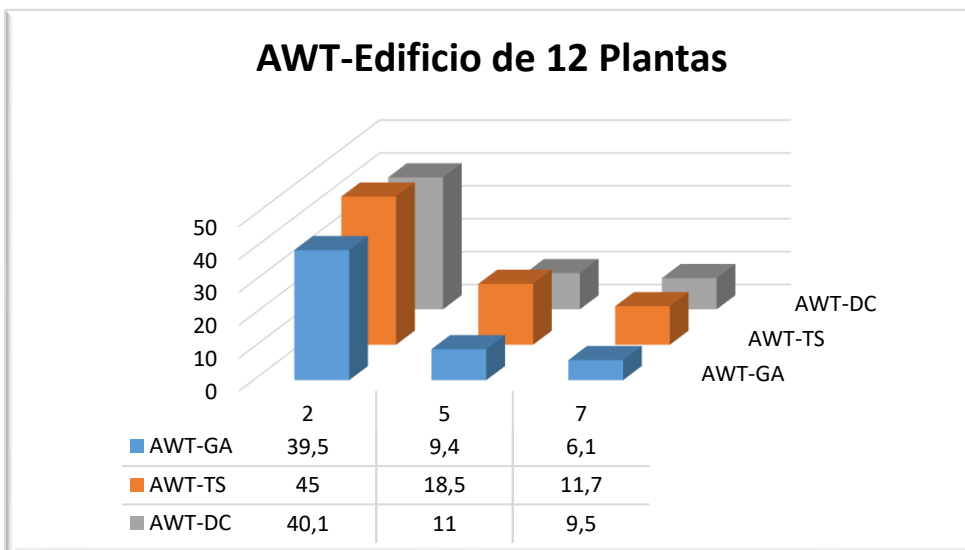


Figura 115: AWT (función de evaluación: tiempo)- 12 Plantas. Lunchpeak

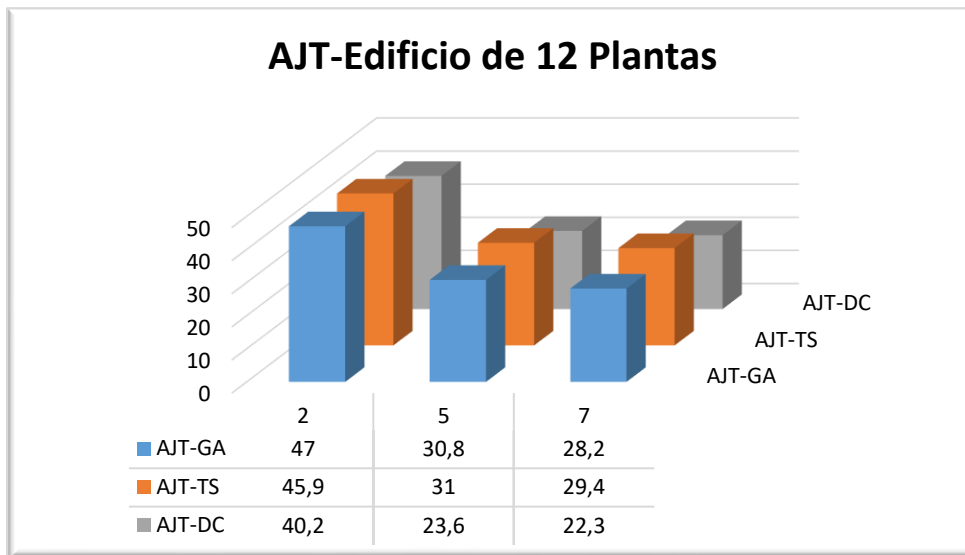


Figura 116: AJT (función de evaluación: tiempo)- 12 Plantas. Lunchpeak

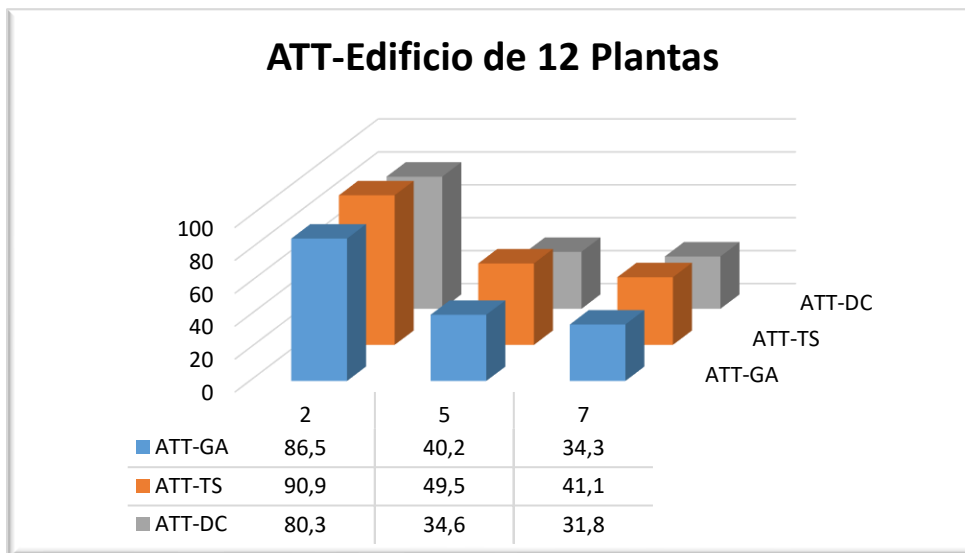


Figura 117: ATT (función de evaluación: tiempo)- 12 Plantas. Lunchpeak

Se observa en la Figura 115 que el algoritmo genético obtiene los mejores tiempos medios de espera. El algoritmo DC-Elevate, seguido del algoritmo genético, obtiene los mejores tiempos medios de tránsito. Para el tiempo medio total se tiene que el algoritmo genético obtiene los mejores resultados junto con el algoritmo implementado por *Elevate*.

Edificio de 20 plantas

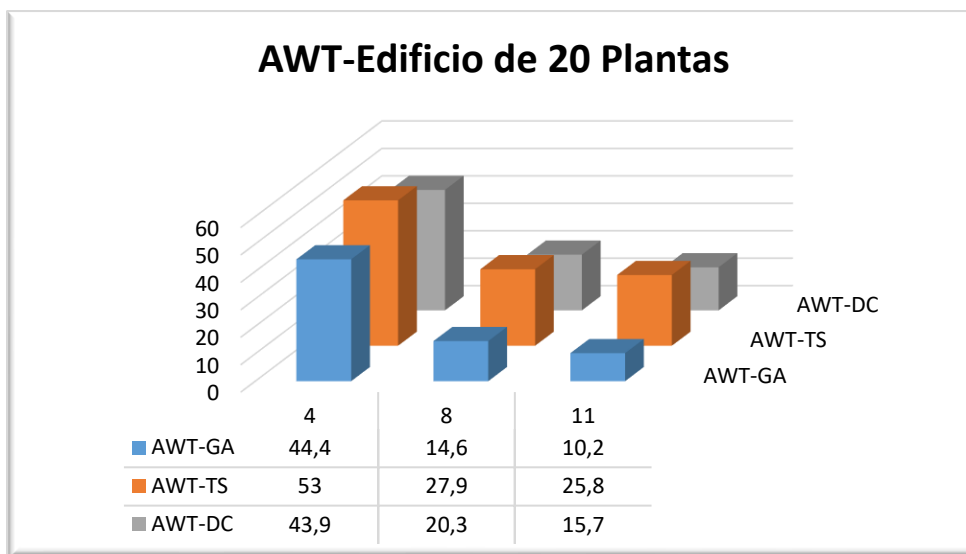


Figura 118: AWT (función de evaluación: tiempo)- 20 Plantas. Lunchpeak

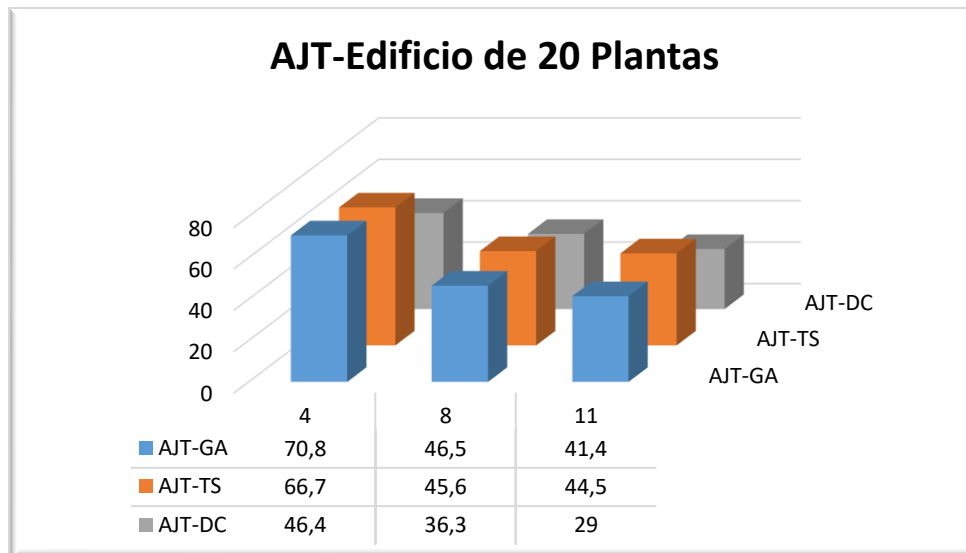


Figura 119: AJT (función de evaluación: tiempo)- 20 Plantas. Lunchpeak

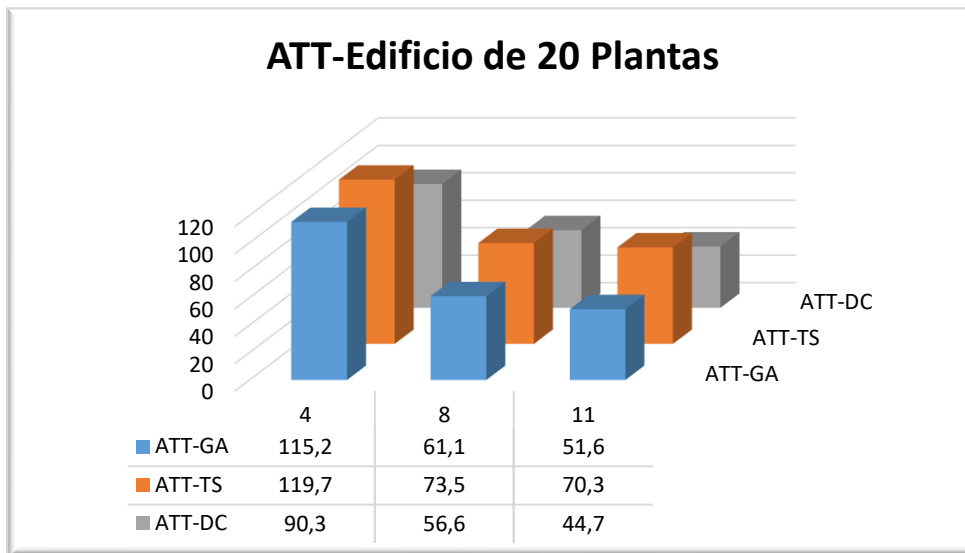


Figura 120: ATT (función de evaluación: tiempo)- 20 Plantas. Lunchpeak

La Figura 118 muestra que el algoritmo genético obtiene los mejores AWT frente al algoritmo DC-Elevate que obtiene los mejores AJT. Finalmente, se tiene que para el tiempo medio total se obtienen los mejores resultados mediante la implementación del algoritmo genético y el implementado por *Elevate*.

Edificio de 24 plantas

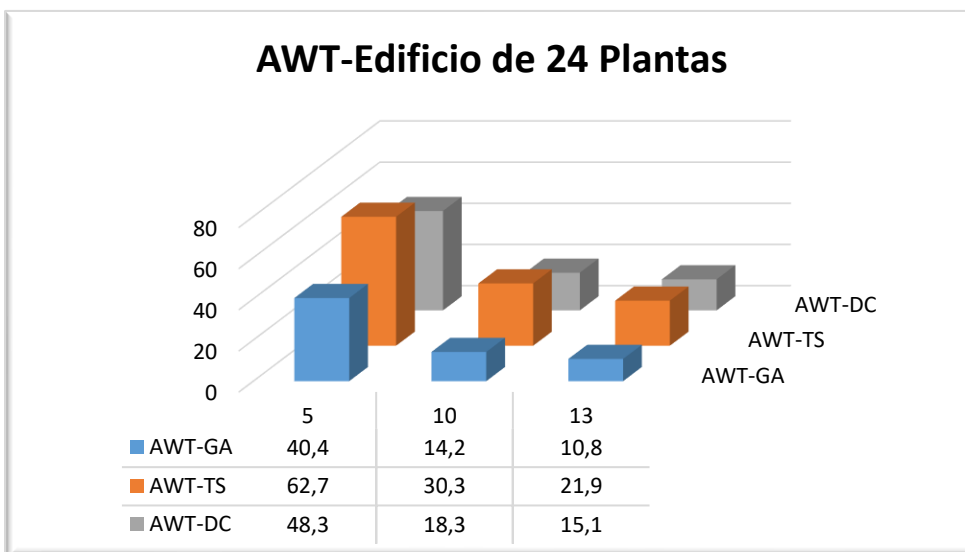


Figura 121: AWT (función de evaluación: tiempo)- 24 Plantas. Lunchpeak

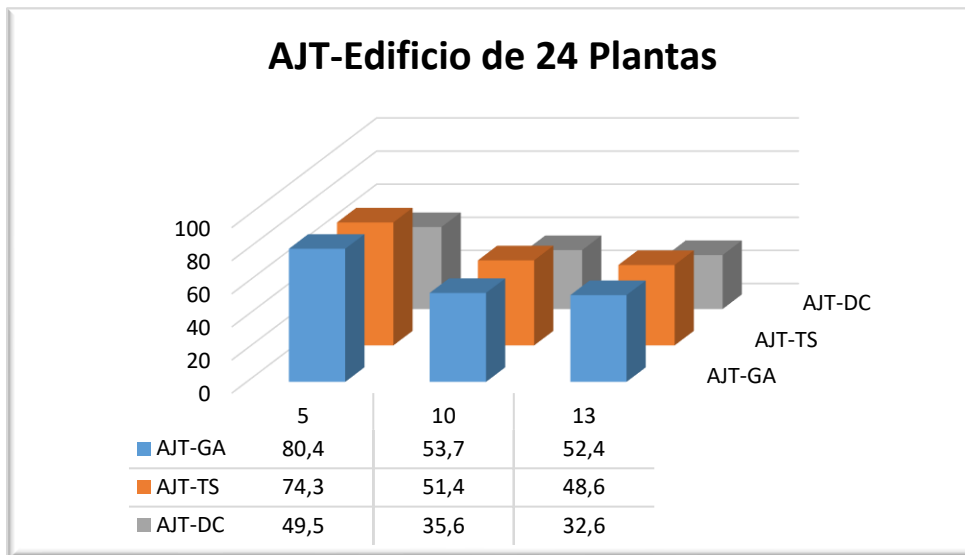


Figura 122: AJT (función de evaluación: tiempo)- 24 Plantas. Lunchpeak

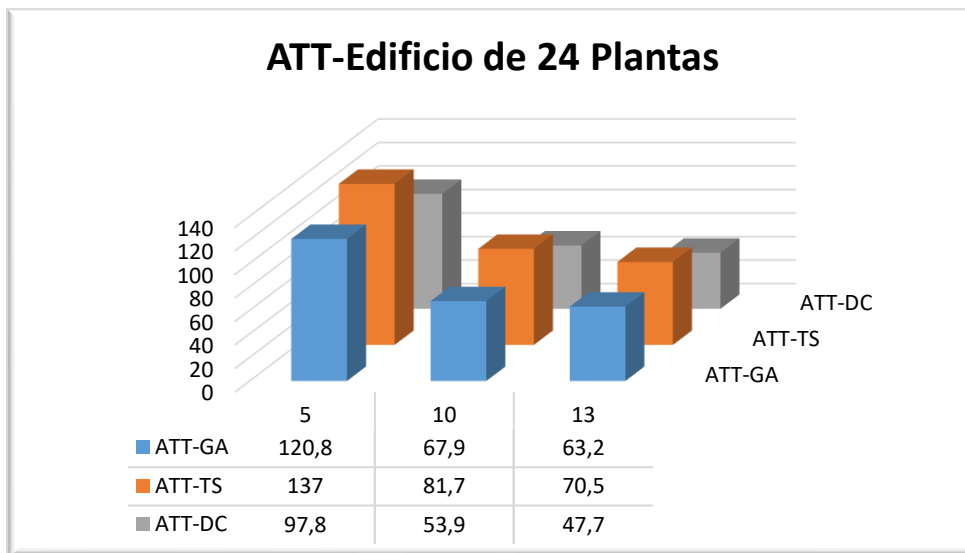


Figura 123: ATT (función de evaluación: tiempo)- 24 Plantas. Lunchpeak

Para un edificio de 24 plantas se observa que el algoritmo genético obtiene los mejores tiempos medios de espera. La Figura 122 refleja que el tiempo medio de tránsito con mejor valor se obtiene mediante el uso de DC-Elevate seguido de la búsqueda tabú. El algoritmo genético junto con el algoritmo implementado por *Elevate* presentan los mejores tiempos totales medios.

Edificio de 28 plantas

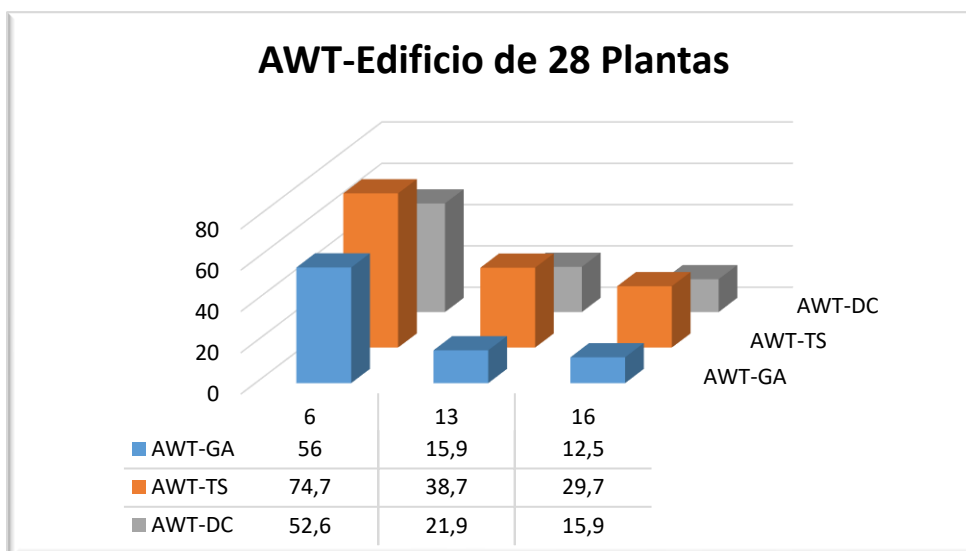


Figura 124: AWT (función de evaluación: tiempo)- 28 Plantas. Lunchpeak

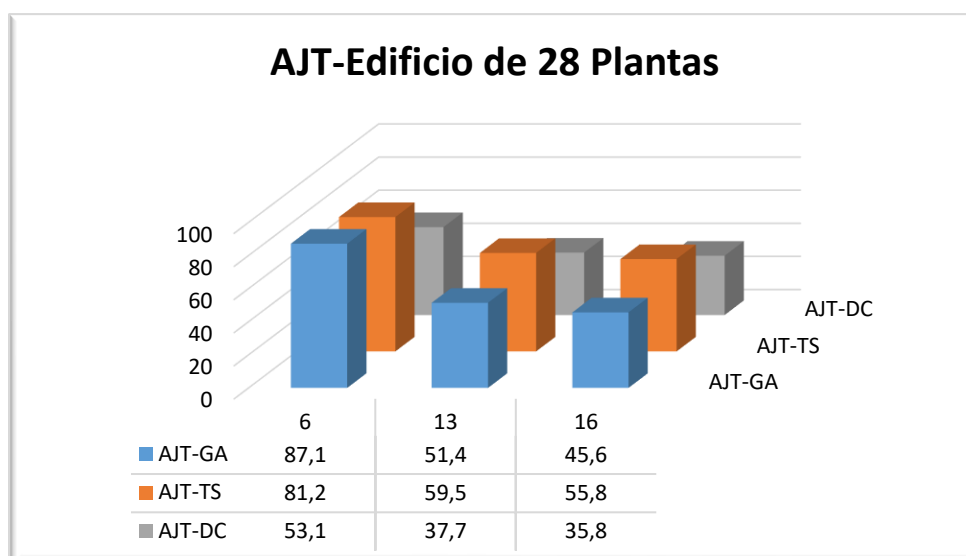


Figura 125: AJT (función de evaluación: tiempo)- 28 Plantas. Lunchpeak

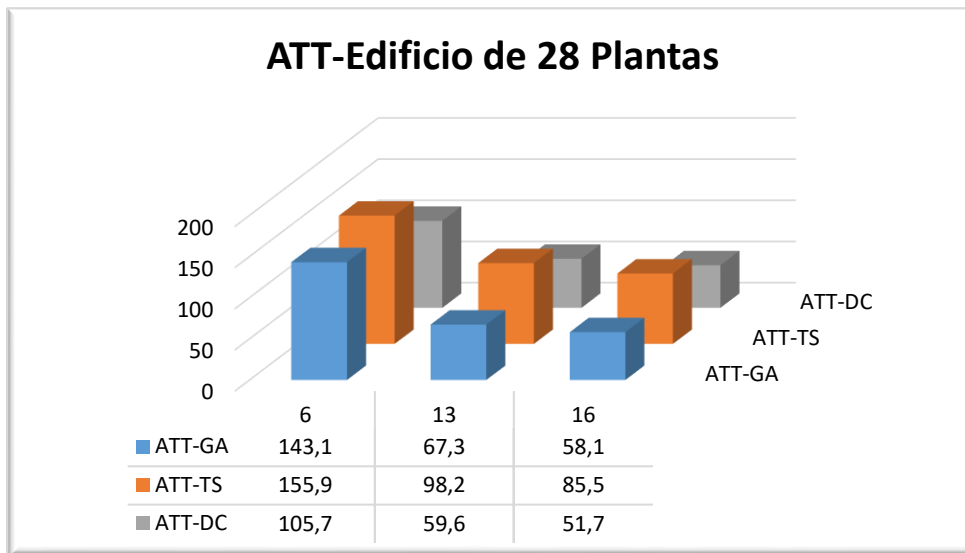


Figura 126: ATT (función de evaluación: tiempo)- 28 Plantas. Lunchpeak

La Figura 124 muestra como el mejor tiempo medio de espera se logra mediante la implementación del algoritmo genético. Para el tiempo medio de tránsito se tiene que el algoritmo DC-Elevate obtiene los mejores tiempos. El algoritmo genético, junto con DC-Elevate, presentan los mejores tiempos medios totales según la Figura 126.

Edificio de 32 plantas

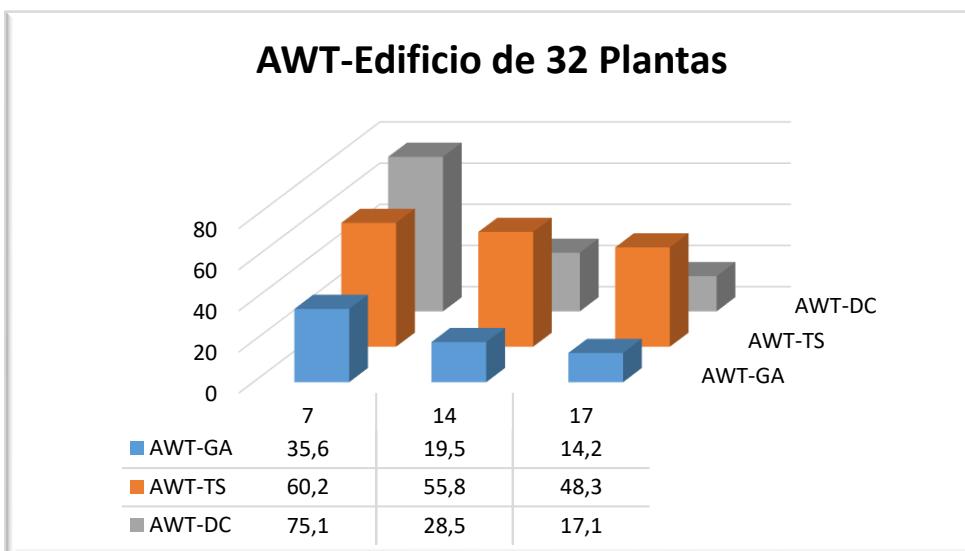


Figura 127: AWT (función de evaluación: tiempo)- 32 Plantas. Lunchpeak

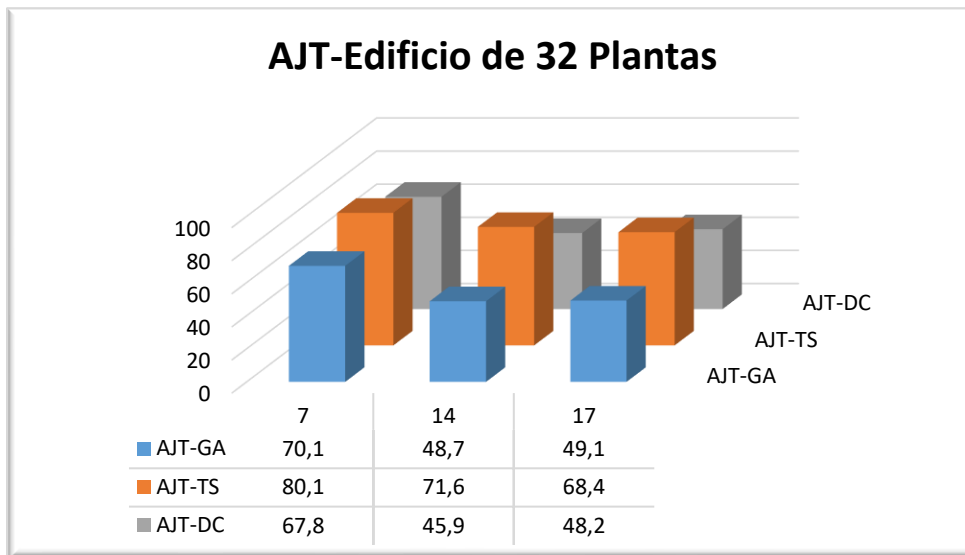


Figura 128: AJT (función de evaluación: tiempo)- 32 Plantas. Lunchpeak

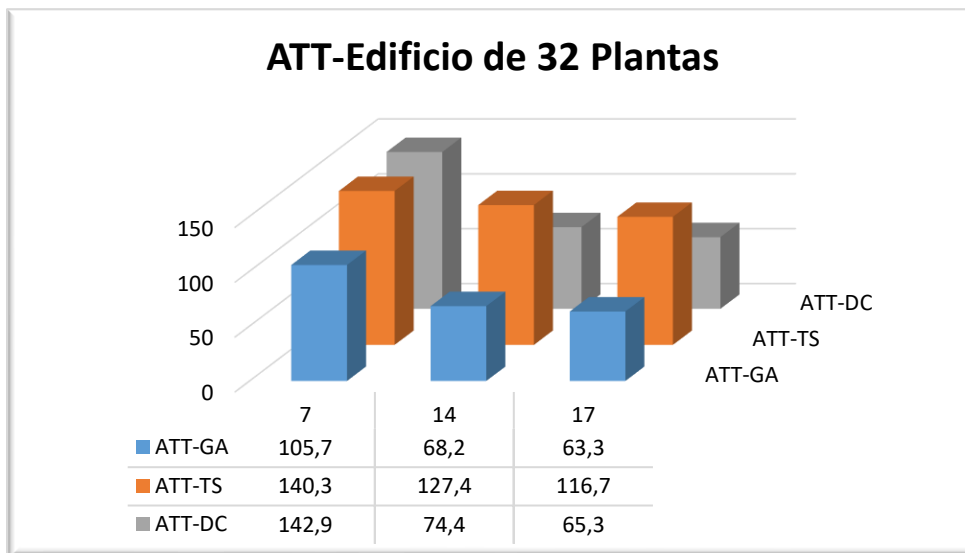


Figura 129: ATT (función de evaluación: tiempo)- 32 Plantas. Lunchpeak

Se muestra en la Figura 127 que el tiempo medio de espera de pasajeros es minimizado por el algoritmo genético para un edificio de 32 plantas. Para el tiempo medio de tránsito existe un empate entre el algoritmo genético y el algoritmo implementado por Elevate. La Figura 129 refleja que el mejor tiempo medio total lo consigue el algoritmo genético.

Edificio de 36 plantas

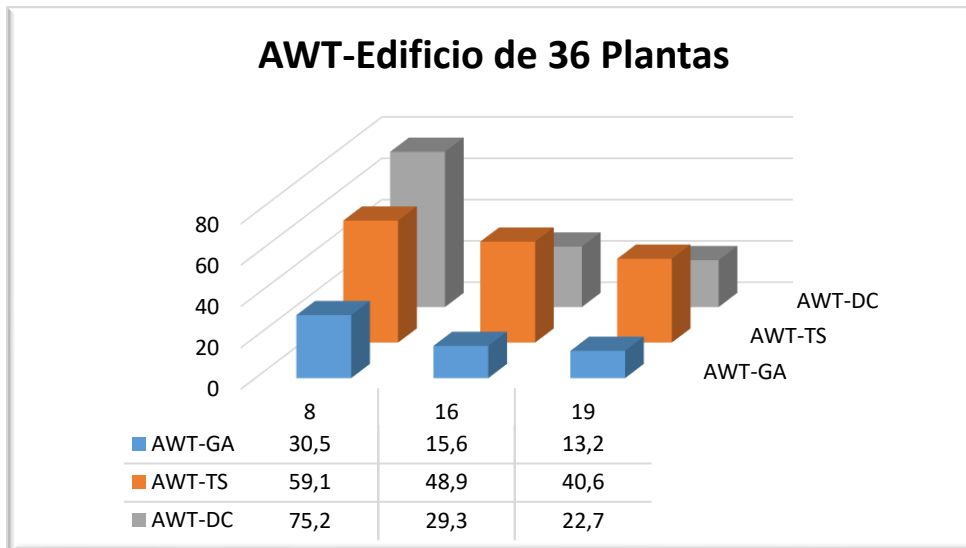


Figura 130: AWT (función de evaluación: tiempo)- 36 Plantas. Lunchpeak

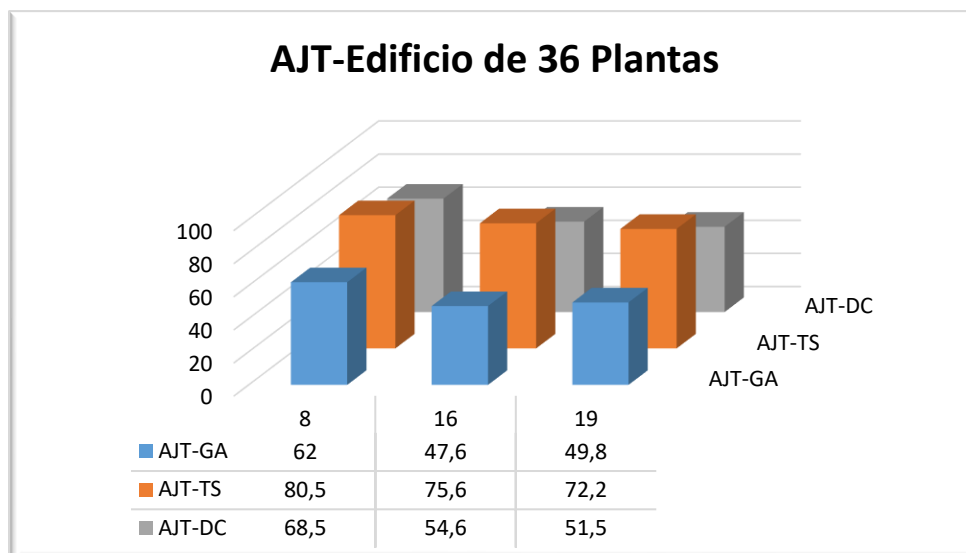


Figura 131: AJT (función de evaluación: tiempo)- 36 Plantas. Lunchpeak

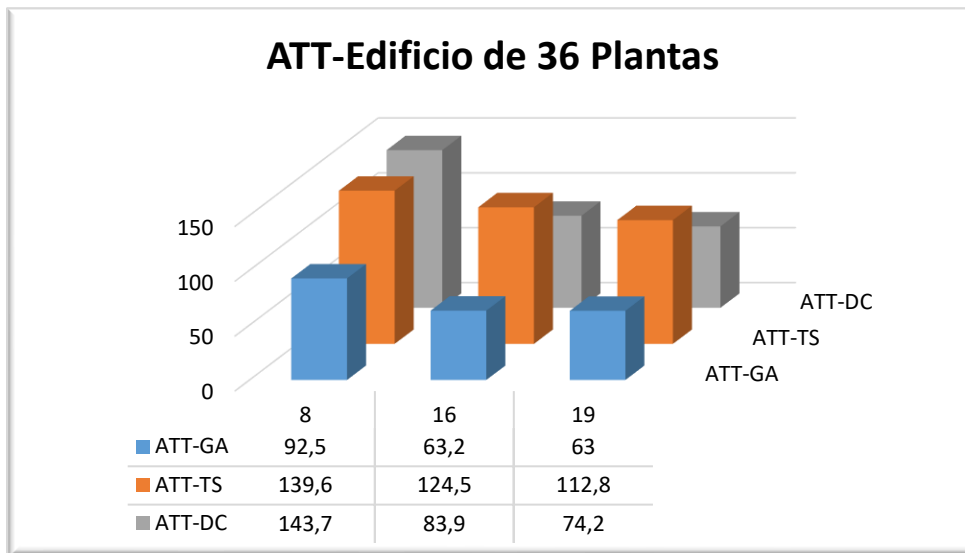


Figura 132: ATT (función de evaluación: tiempo)- 36 Plantas. Lunchpeak

Para un edificio de 36 plantas se observa en la Figura 130 que el AWT se minimiza mediante la implementación del algoritmo genético. El tiempo medio de tránsito obtiene los mejores valores para el algoritmo genético. Por tanto, de forma evidente pues el tiempo medio total resulta de la suma de los dos tiempos anteriores, este tiempo minimizado lo obtiene nuevamente el algoritmo genético según la Figura 132.

Edificio de 40 plantas

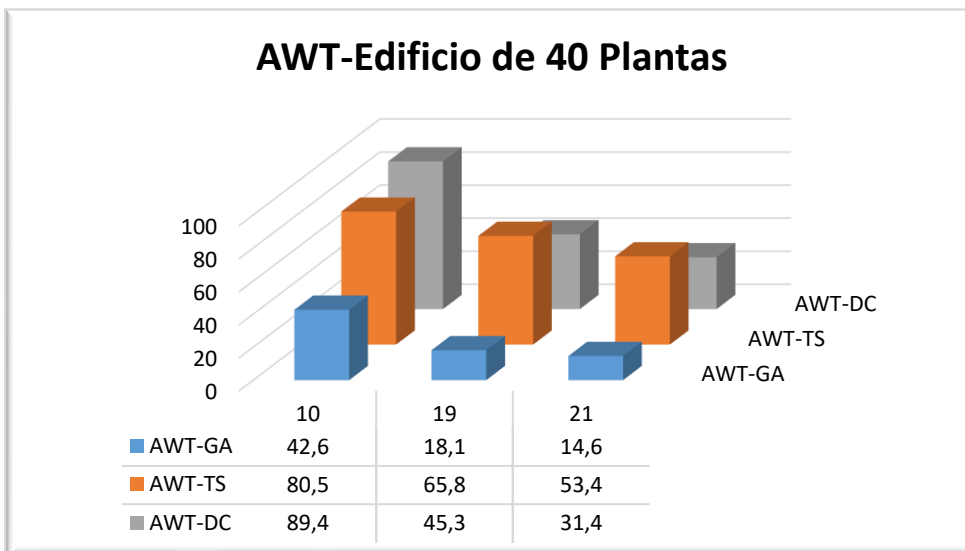


Figura 133: AWT (función de evaluación: tiempo)- 40 Plantas. Lunchpeak

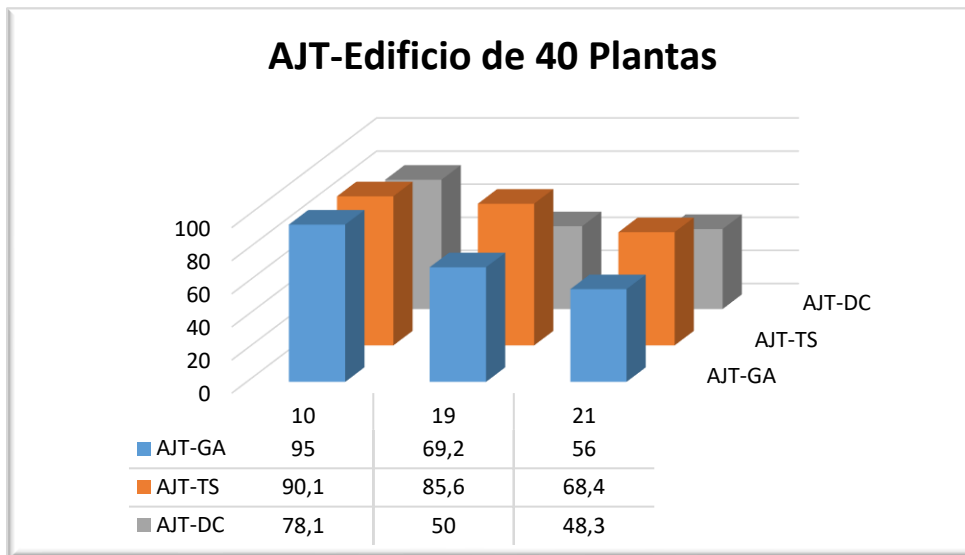


Figura 134: AJT (función de evaluación: tiempo)- 40 Plantas. Lunchpeak

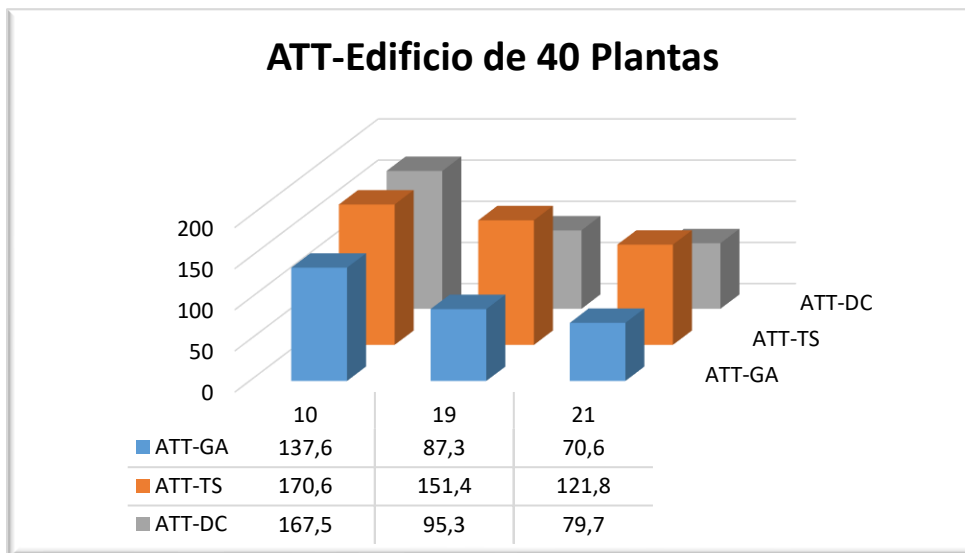


Figura 135: ATT (función de evaluación: tiempo)- 40 Plantas. Lunchpeak

En un edificio de 40 plantas se tiene que el algoritmo genético obtiene los mejores tiempos medios de espera mientras que la Figura 134 refleja que DC-Elevate consigue los mejores tiempos medios de tránsito. Para el tiempo medio total se observa en la Figura 135 que tanto el algoritmo genético como el implementado por *Elevate* obtienen los mejores tiempos medios totales.

6.2.3. Tráfico Downpeak

Edificio de 12 plantas

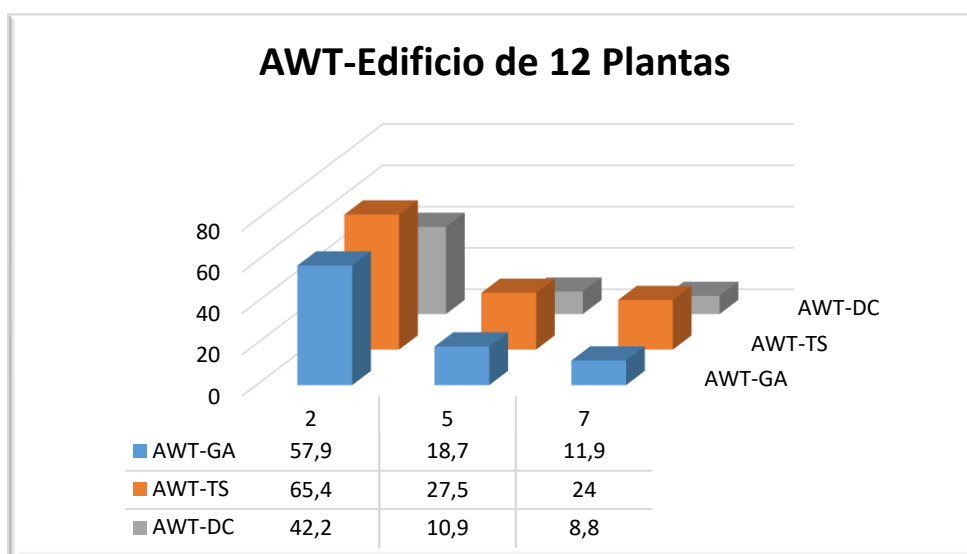


Figura 136: AWT (función de evaluación: tiempo)- 12 Plantas. Downpeak

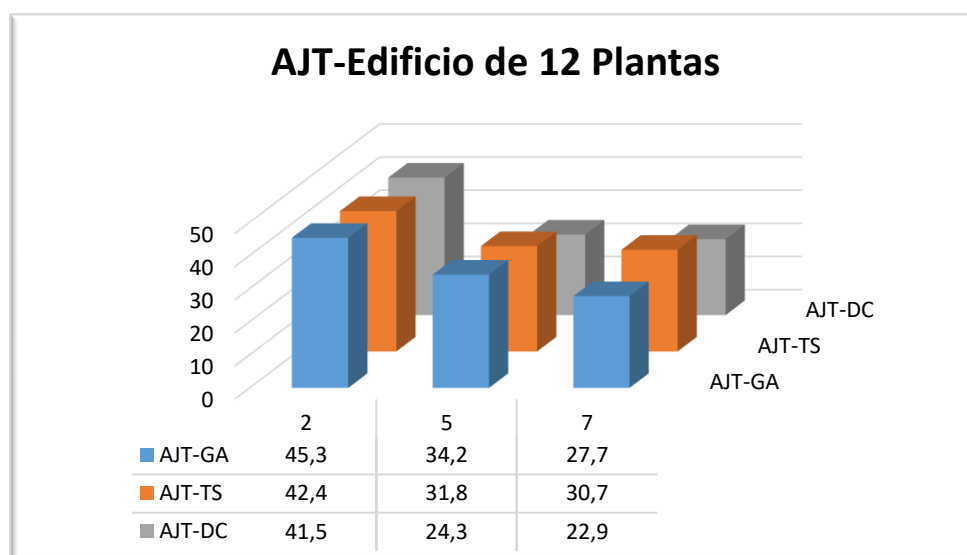


Figura 137: AJT (función de evaluación: tiempo)- 12 Plantas. Downpeak

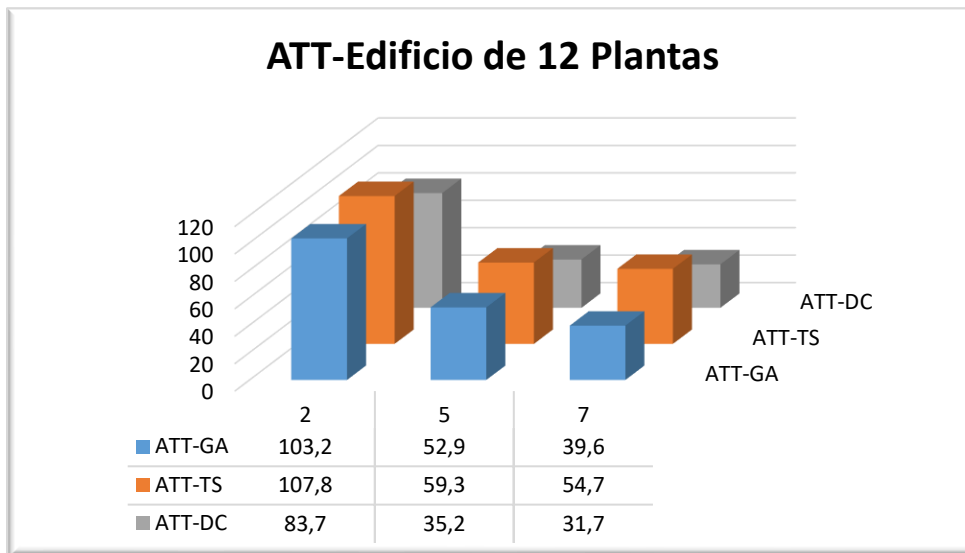


Figura 138: ATT (función de evaluación: tiempo)- 12 Plantas. Downpeak

Para un edificio de 12 plantas se observa en la Figura 136 que DC-Elevate arroja los mejores tiempos medios de espera, seguido del algoritmo genético. Para el tiempo medio de tránsito se tiene que los tres algoritmos obtienen unos tiempos parecidos. En cambio, para el tiempo medio total se obtiene que el algoritmo DC-Elevate junto con el algoritmo genético obtienen los mejores resultados.

Edificio de 20 plantas

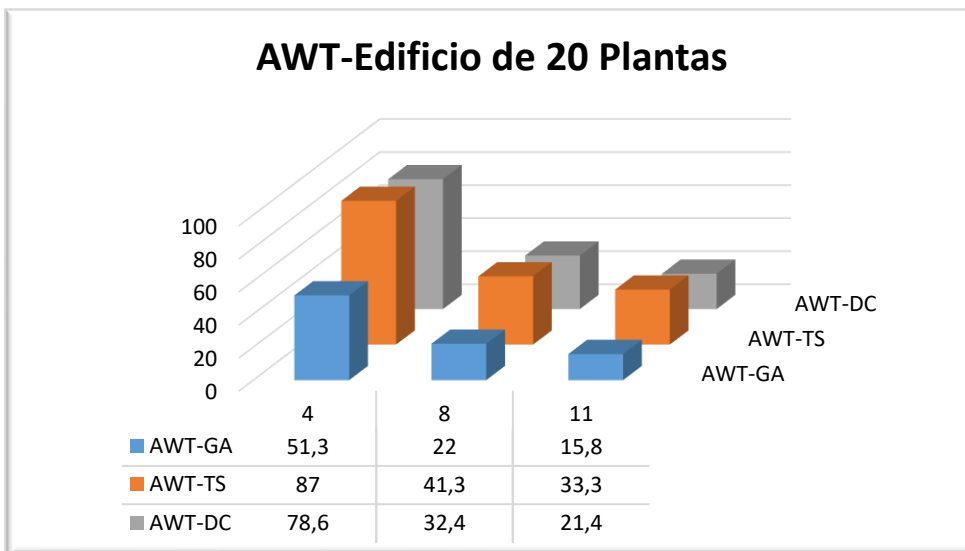


Figura 139: AWT (función de evaluación: tiempo)- 20 Plantas. Downpeak

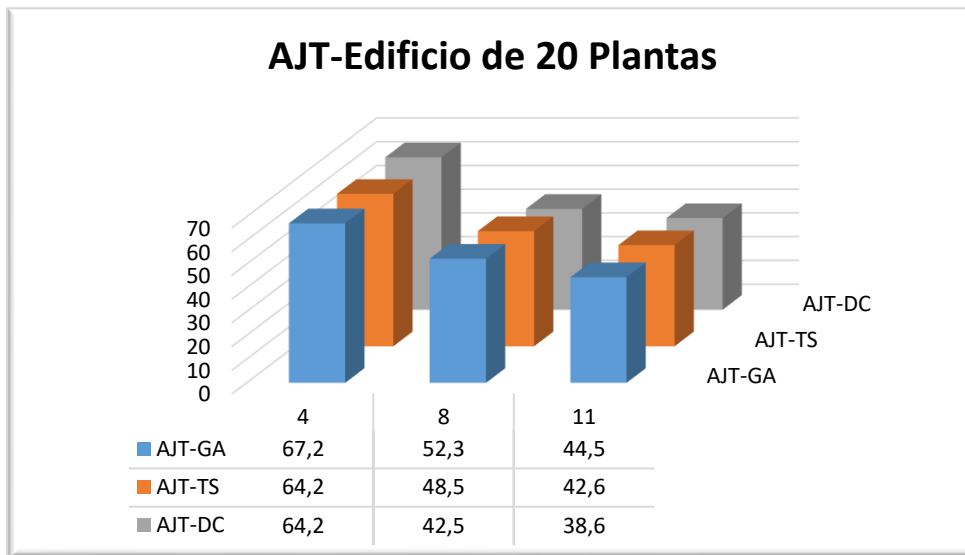


Figura 140: AJT (función de evaluación: tiempo)- 20 Plantas. Downpeak

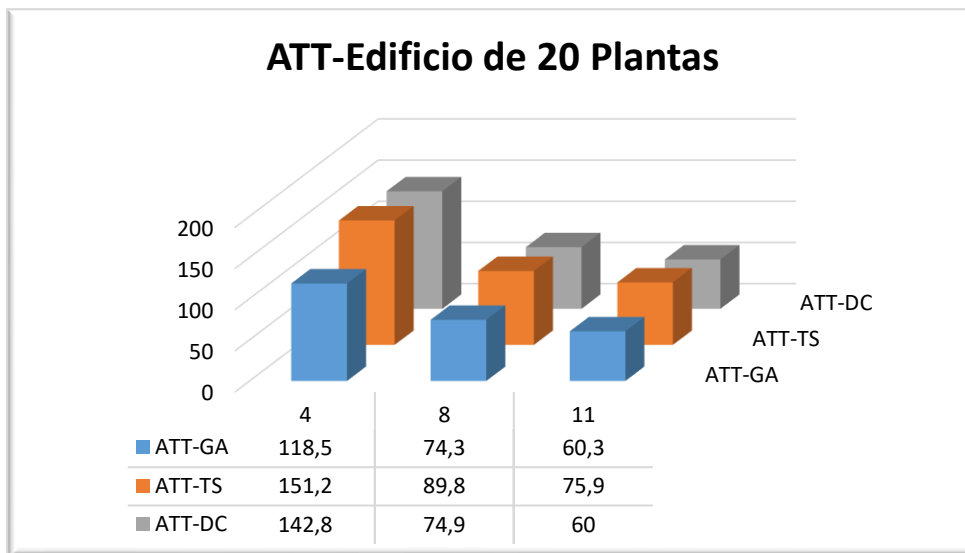


Figura 141: ATT (función de evaluación: tiempo)- 20 Plantas. Downpeak

La Figura 139 refleja que el tiempo medio de espera lo consigue minimizar el algoritmo genético para un edificio de 20 plantas. Se obtiene un triple empate entre los tres algoritmos para el tiempo medio de tránsito. Para el tiempo medio total, la implementación del algoritmo genético o del algoritmo DC-Elevate consigue los mejores tiempos medios totales según la Figura 141.

Edificio de 24 plantas

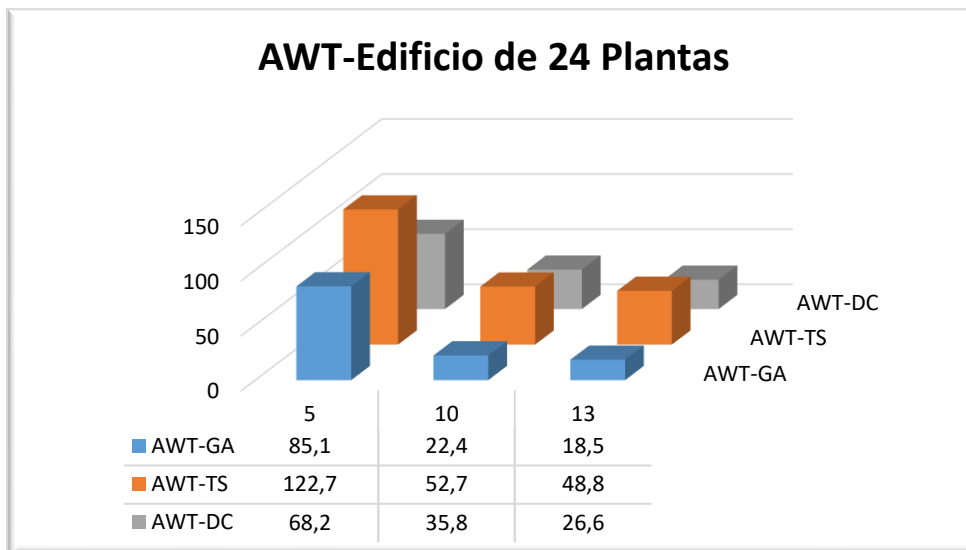


Figura 142: AWT (función de evaluación: tiempo)- 24 Plantas. Downpeak

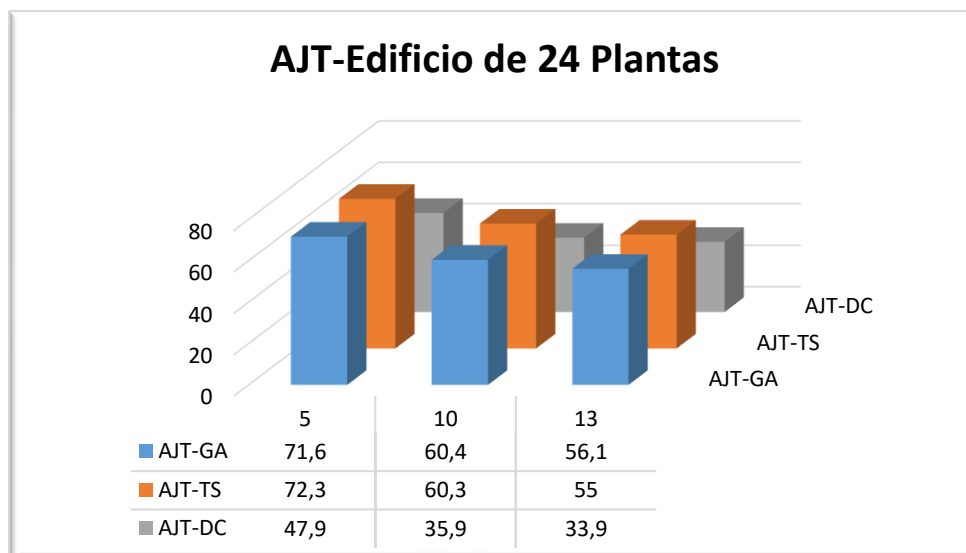


Figura 143: AJT (función de evaluación: tiempo)- 24 Plantas. Downpeak

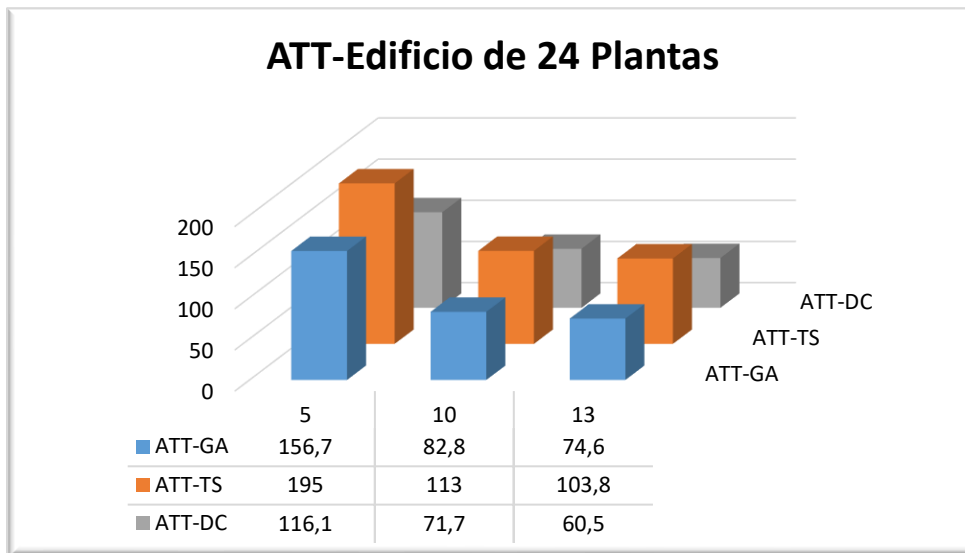


Figura 144: ATT (función de evaluación: tiempo)- 24 Plantas. Downpeak

Para un edificio de 24 plantas se aprecia en la Figura 142 que el algoritmo genético obtiene los mejores tiempos medios de espera. El tiempo medio de tránsito es minimizado mediante DC-Elevate. Se observa en la Figura 144 que el algoritmo genético junto con el implementado por *Elevate* obtienen los mejores tiempos medios totales.

Edificio de 28 plantas

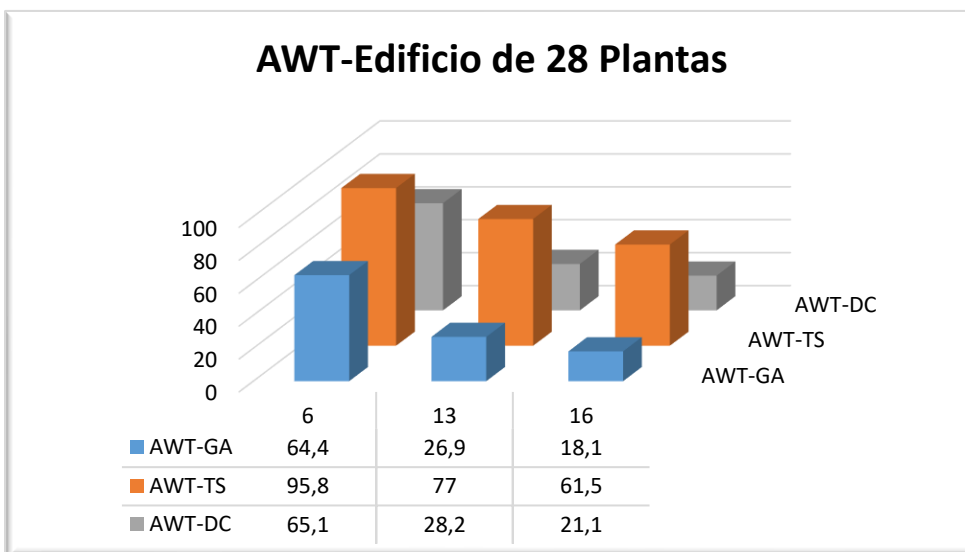


Figura 145: AWT (función de evaluación: tiempo)- 28 Plantas. Downpeak

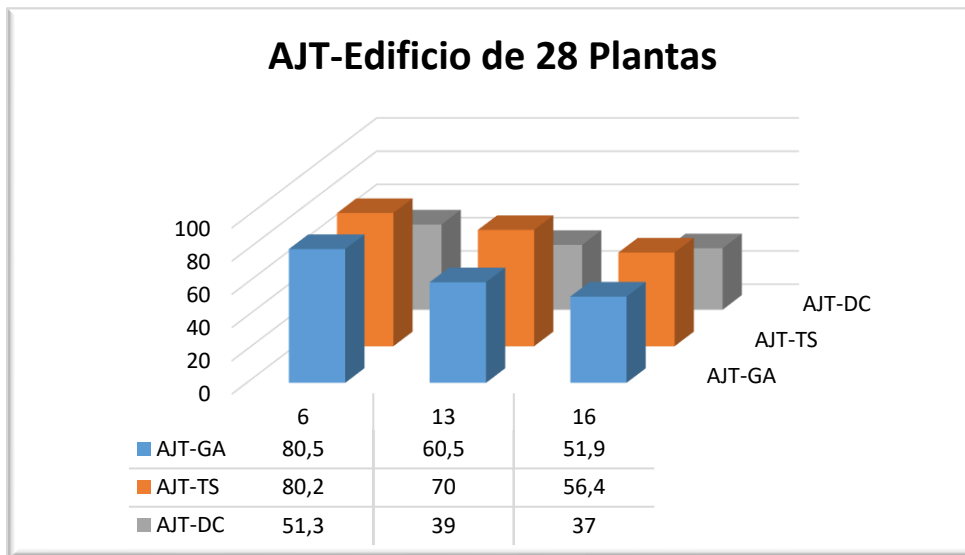


Figura 146: AJT (función de evaluación: tiempo)- 28 Plantas. Downpeak

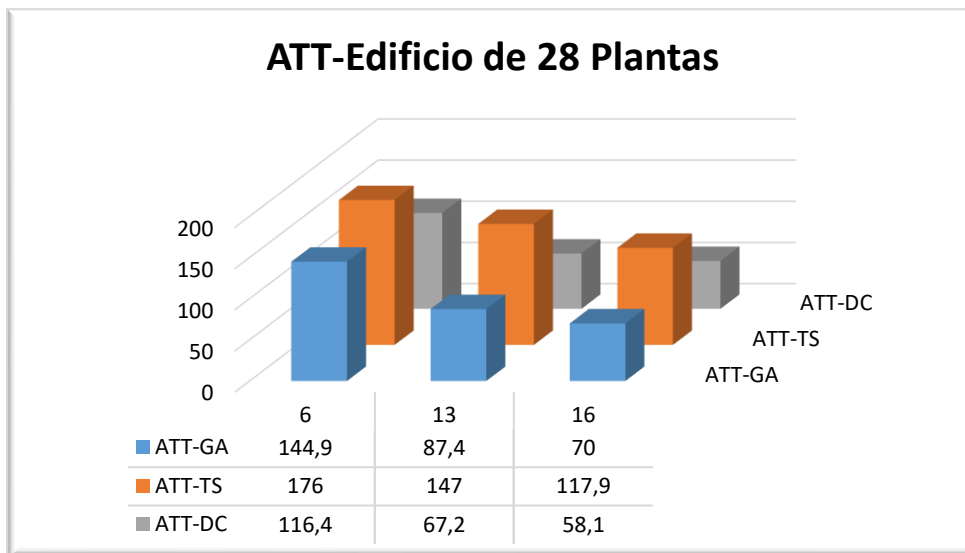


Figura 147: ATT (función de evaluación: tiempo)- 28 Plantas. Downpeak

La Figura 145 refleja que para un edificio de 28 plantas el tiempo medio de espera lo minimiza el algoritmo genético. El tiempo medio de tránsito se consigue minimizar mediante DC-Elevate. El algoritmo genético junto con DC-Elevate obtienen los mejores tiempos medios totales según se observa en la Figura 147.

Edificio de 32 plantas

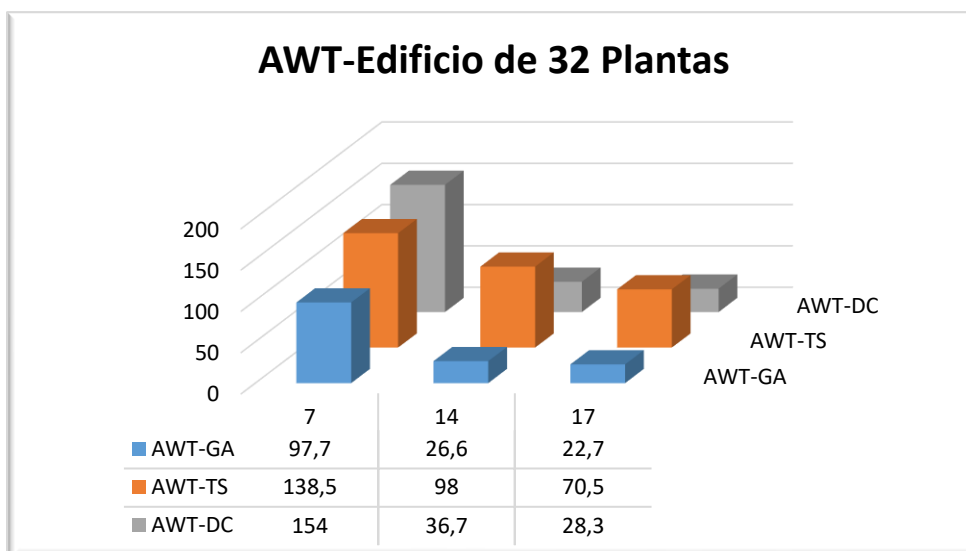


Figura 148: AWT (función de evaluación: tiempo)- 32 Plantas. Downpeak

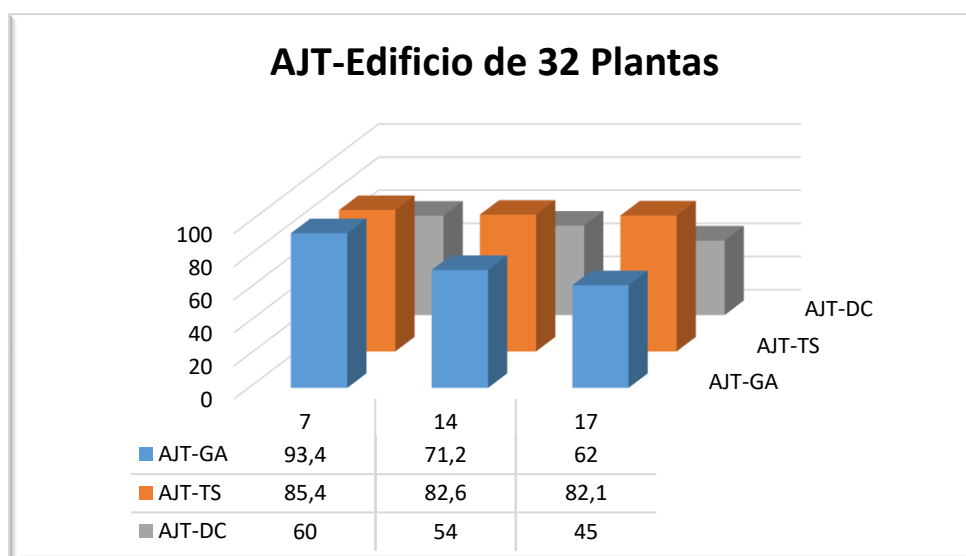


Figura 149: AJT (función de evaluación: tiempo)- 32 Plantas. Downpeak

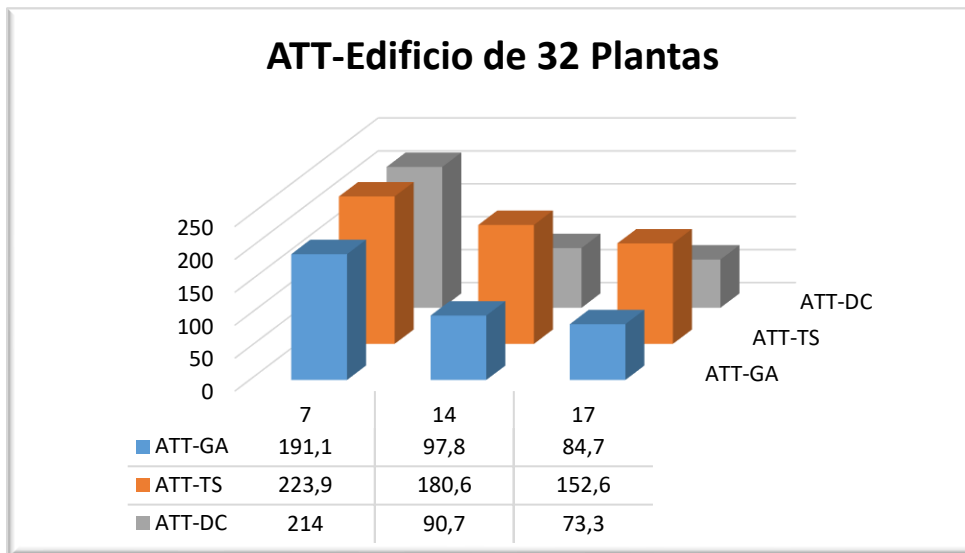


Figura 150: ATT (función de evaluación: tiempo)- 32 Plantas. Downpeak

Para un edificio de 32 plantas se observa en la Figura 148 que el algoritmo genético minimiza el tiempo medio de espera. DC-Elevate es el algoritmo que consigue la minimización el tiempo medio de tránsito frente al algoritmo genético que consigue junto al algoritmo implementado por *Elevate* la mejora del tiempo medio total.

Edificio de 36 plantas

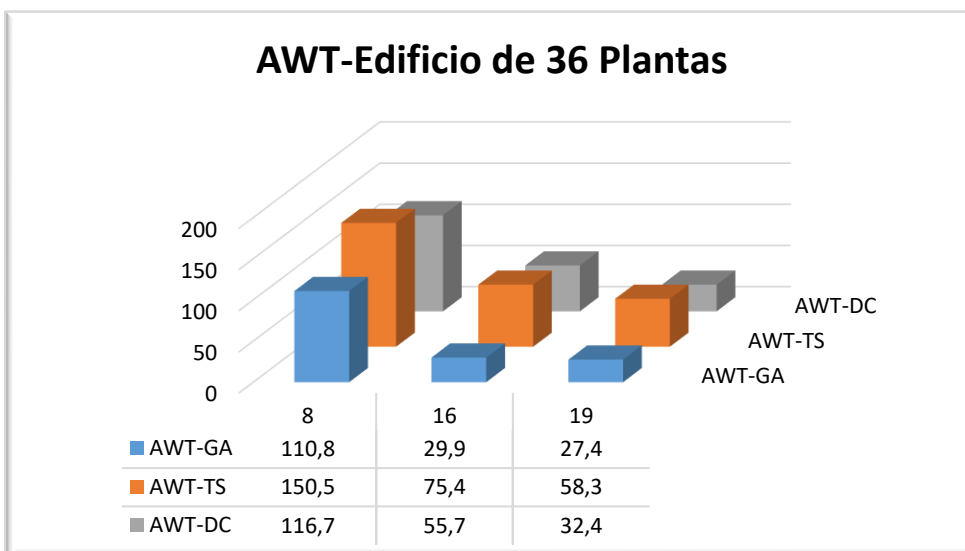


Figura 151: AWT (función de evaluación: tiempo)- 36 Plantas. Downpeak

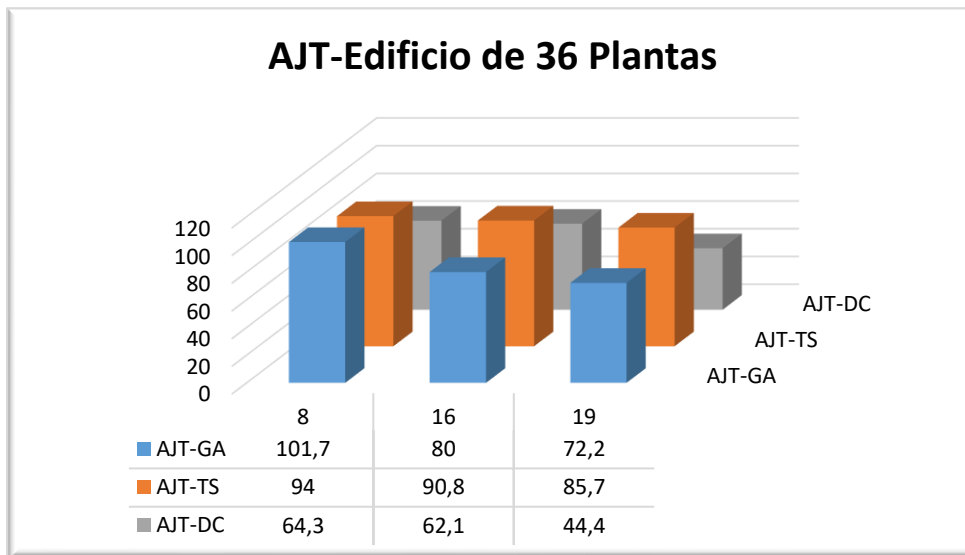


Figura 152: AJT (función de evaluación: tiempo)- 36 Plantas. Downpeak

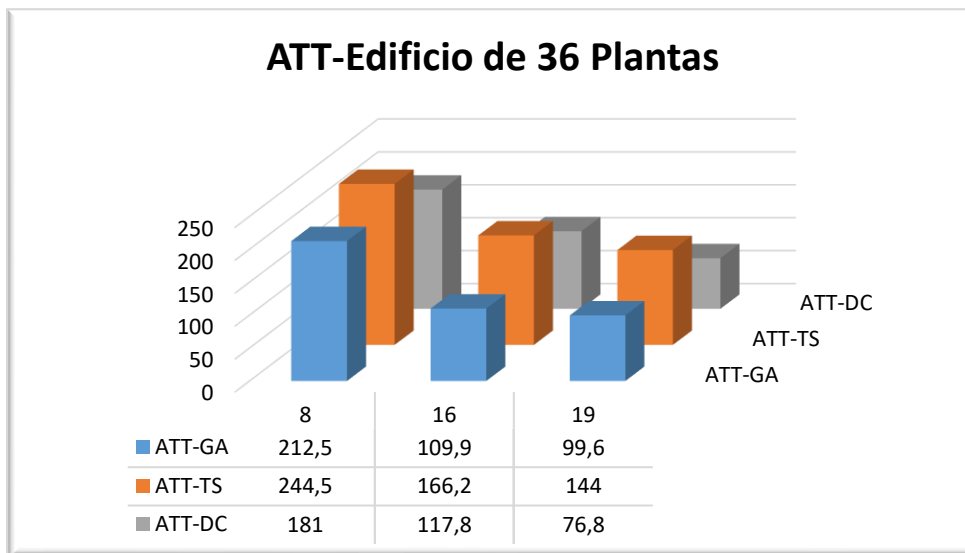


Figura 153: ATT (función de evaluación: tiempo)- 36 Plantas. Downpeak

En un edificio de 36 plantas se tiene que el algoritmo genético consigue la minimización del tiempo medio de espera. El AJT se minimiza mediante el algoritmo DC-Elevate como se observa en la Figura 152. Es el algoritmo genético el que obtiene los mejores tiempos medios totales junto con el algoritmo implementado por *Elevate*.

Edificio de 40 plantas

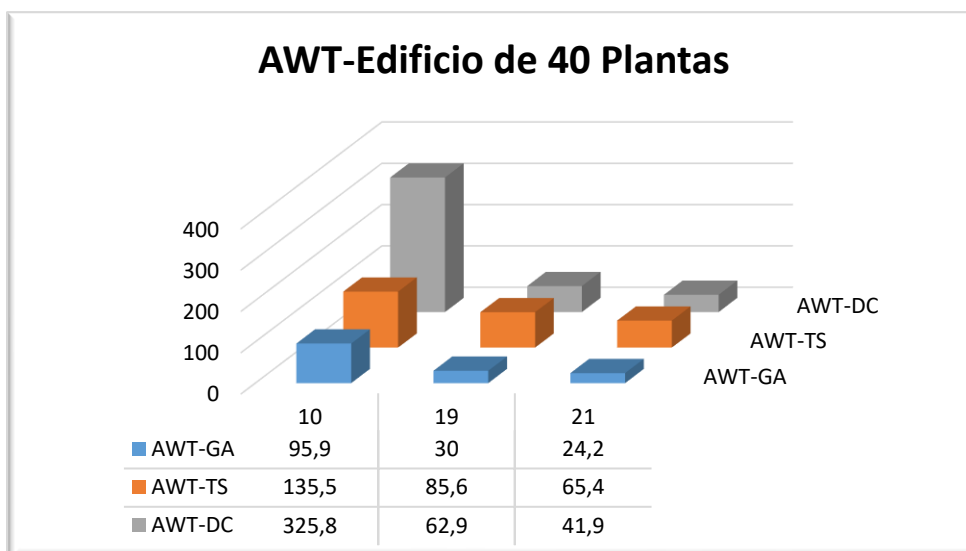


Figura 154: AWT (función de evaluación: tiempo)- 40 Plantas. Downpeak

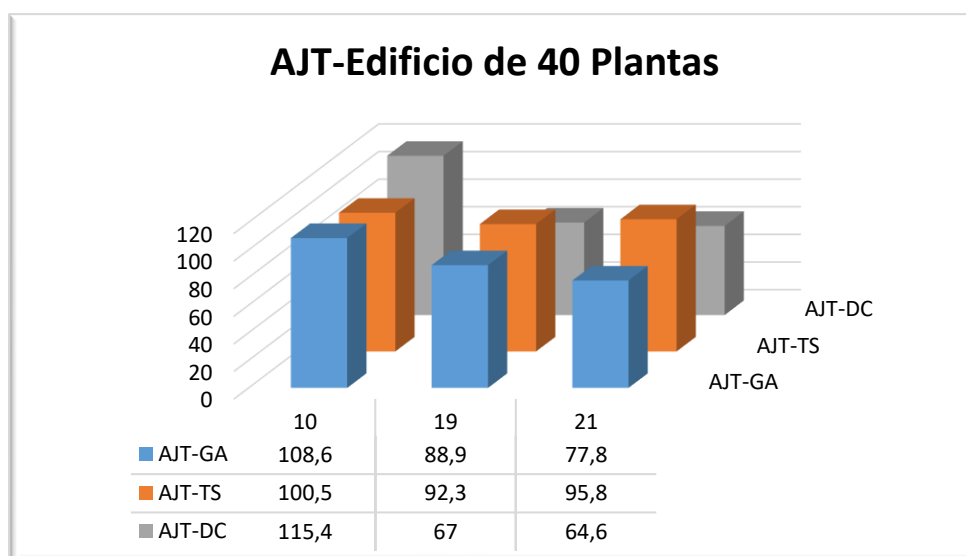


Figura 155: AJT (función de evaluación: tiempo)- 40 Plantas. Downpeak

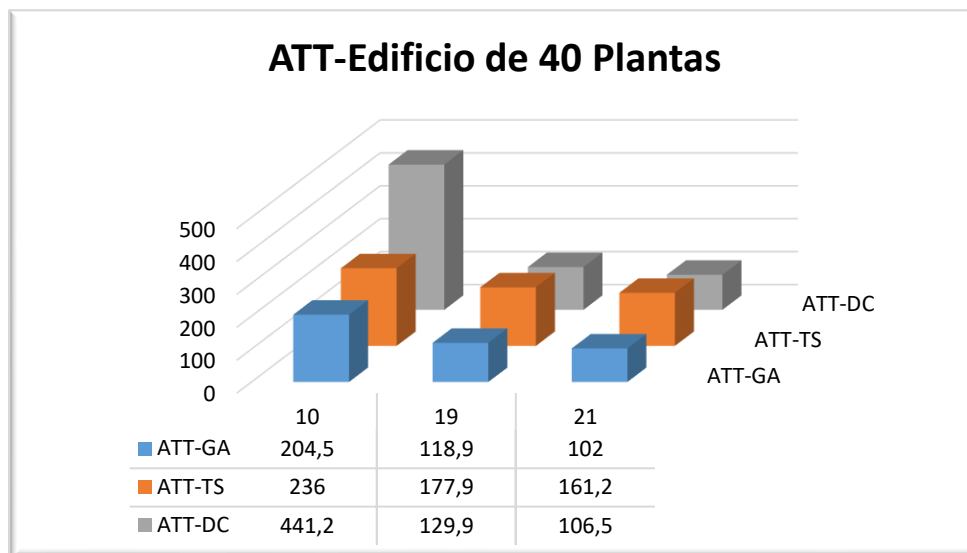


Figura 156: ATT (función de evaluación: tiempo)- 40 Plantas. Downpeak

Para un edificio de 40 plantas se observa en la Figura 154 que el algoritmo genético obtiene los mejores tiempos medios de espera. Para el tiempo medio de tránsito se tiene en la Figura 155 que se minimiza mediante DC-Elevate seguido del algoritmo genético. La Figura 156 refleja que el algoritmo genético consigue los mejores valores para el tiempo medio total.

7. ANÁLISIS COMPARATIVO DEL CONSUMO DE ENERGÍA

Se presenta en este apartado el estudio de la energía consumida por los ascensores de los edificios en función de la función de evaluación empleada y el método de resolución, así como del patrón de tráfico. Al igual que en el apartado anterior, se presentan el análisis en subapartados de estudio del consumo energético acumulado.

7.1. Tráfico Interfloor

Para el consumo energético del tipo de tráfico Interfloor, se puede apreciar en la Figura 157 que para la función de evaluación basada en distancia el menor consumo lo consigue el algoritmo genético (color azul) seguido de la búsqueda tabú (color naranja). Del mismo modo ocurre para la función basada en tiempo que se muestra en la Figura 158. Se evidencia que el consumo energético de DC-Elevate es muy elevado con respecto a los otros dos algoritmos.

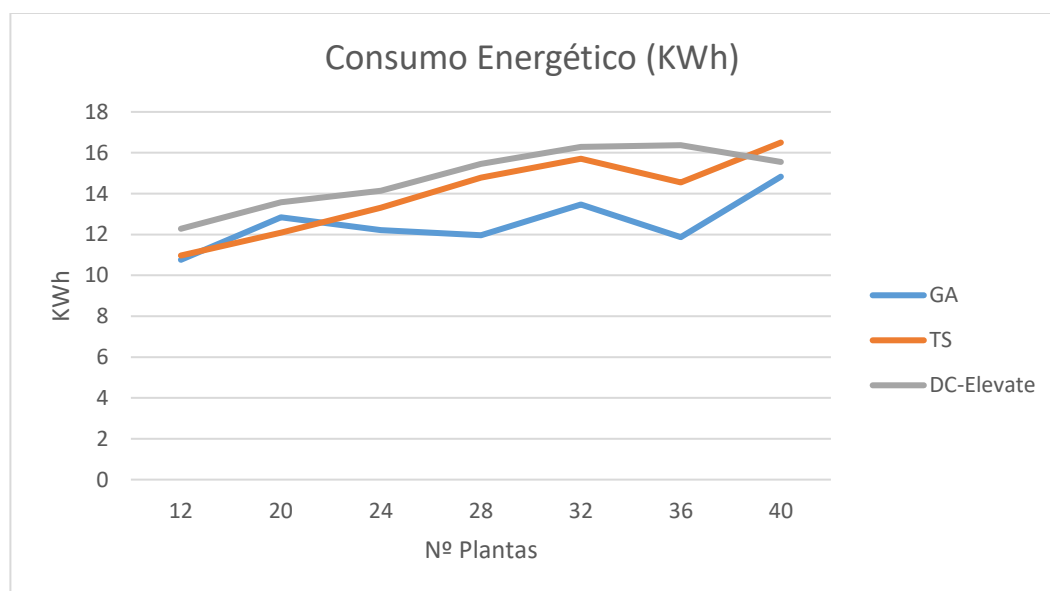


Figura 157: Consumo energético (función de evaluación: distancia). Interfloor

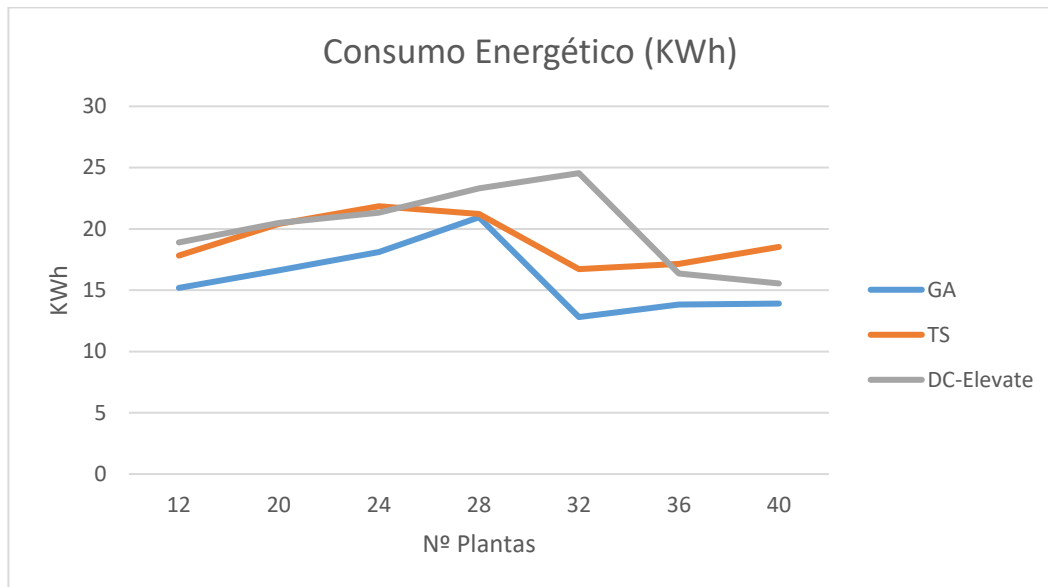


Figura 158: Consumo energético (función de evaluación: tiempo). Interfloor

7.2. Tráfico Lunchpeak

El consumo energético para el tráfico Lunchpeak refleja en la Figura 159 que para la función de evaluación basada en distancia el algoritmo genético (color azul) y la búsqueda tabú (color naranja) disputan por la obtención del menor consumo. En cambio, para la función de evaluación basada en tiempo es el algoritmo genético el que obtiene los mejores consumos como se observa en la Figura 160. Para ambas funciones de evaluación, se observa como el algoritmo implementado por *Elevate* obtiene los consumos energéticos más altos.

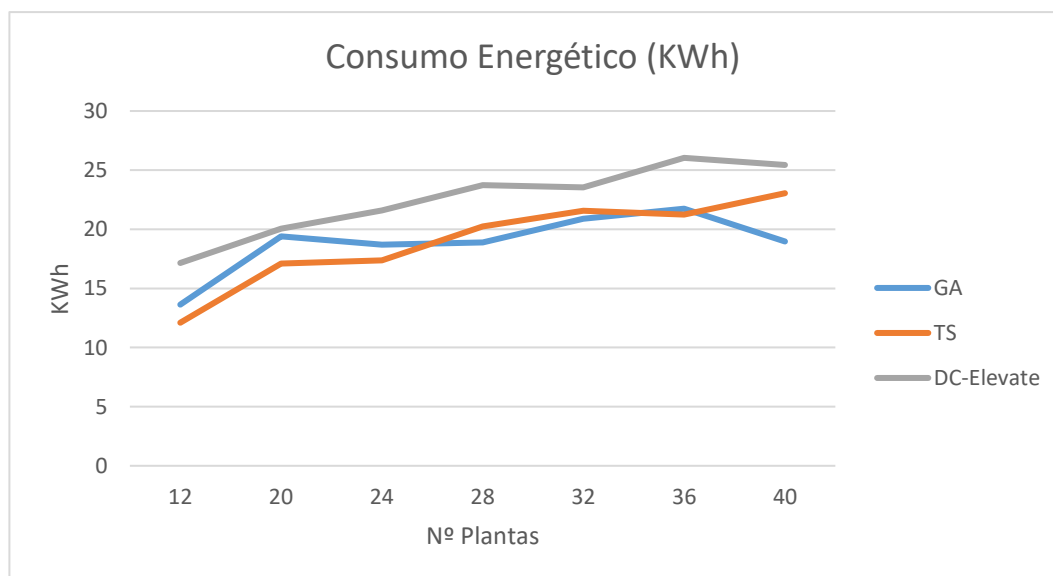


Figura 159: Consumo energético (función de evaluación: distancia). Lunchpeak

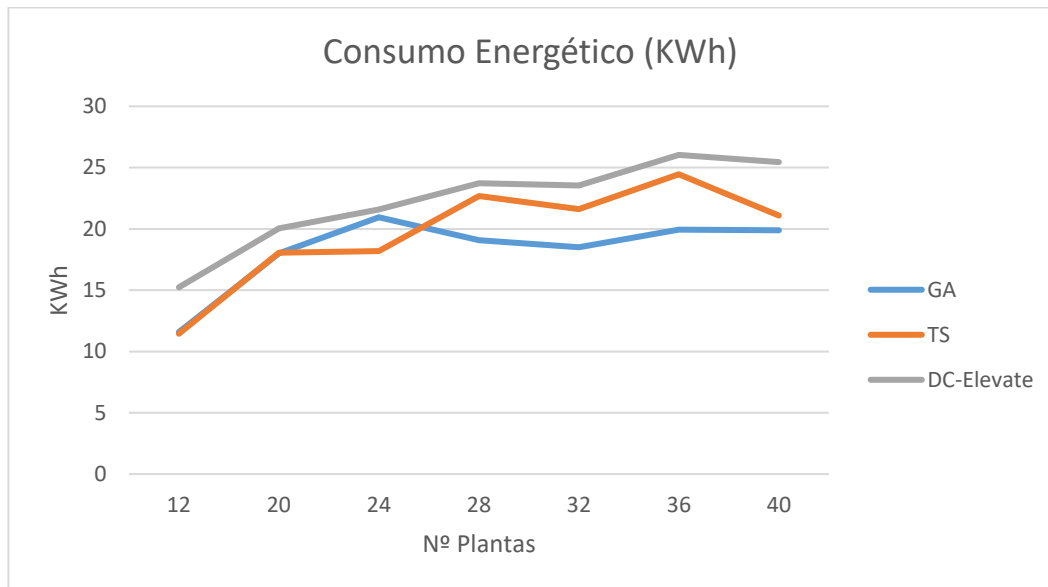


Figura 160: Consumo energético (función de evaluación: tiempo). Lunchpeak

7.3. Tráfico Downpeak

Del estudio del consumo energético se extrae de la Figura 161 y Figura 162 que para el tipo de tráfico Downpeak, tanto el algoritmo genético (color azul) como la búsqueda tabú (color naranja) aportan buenos valores para la minimización del consumo ya sea mediante el empleo de la función de evaluación basada en distancia o en tiempo. De igual modo que en los otros dos tráficos anteriores, se extrae que el algoritmo DC-Elevate obtiene los peores consumos energéticos.

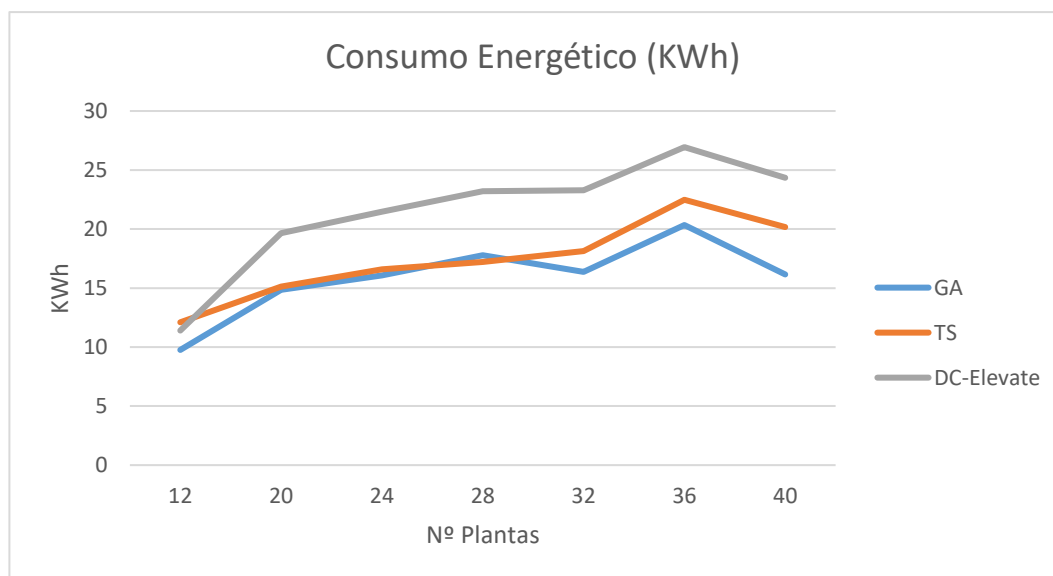


Figura 161: Consumo energético (función de evaluación: distancia). Downpeak

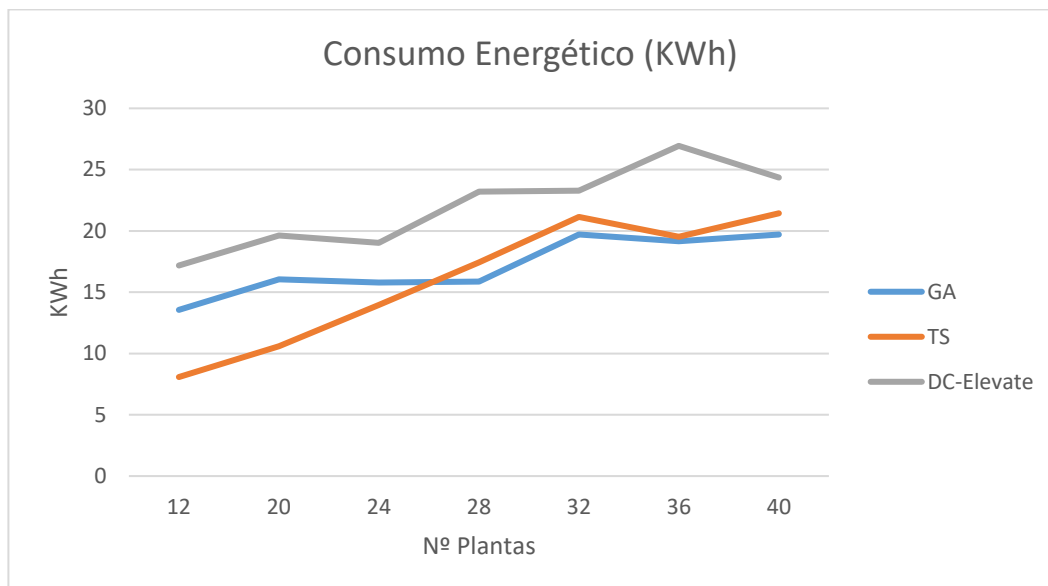


Figura 162: Consumo energético (función de evaluación: tiempo). Downpeak

8. ALGORITMOS MÁS EFICIENTES SEGÚN LAS CARACTERÍSTICAS DEL EDIFICIO

Este apartado engloba el análisis para cada patrón de tráfico de los algoritmos que consiguen unos mejores valores de tiempo medio de espera y tiempo medio de tránsito. Para ello, se emplean unas gráficas 3D generadas mediante el software *Matlab* en las que se representa en los ejes el número de plantas de los edificios, el número de ascensores y el tiempo medio de espera o tránsito. Cada método resolutorio se representa en las gráficas de un color, de modo que aquellos colores que se encuentren más abajo son los que consiguen obtener menores tiempos.

8.1. Tráfico Interfloor

Para el tráfico Interfloor se observa en la Figura 163 y Figura 164 que se obtiene un mejor tiempo medio de espera mediante el algoritmo genético (color rojo) seguido del algoritmo de búsqueda tabú (color verde) empleándose una función objetivo de distancia y tiempo, respectivamente en las figuras.

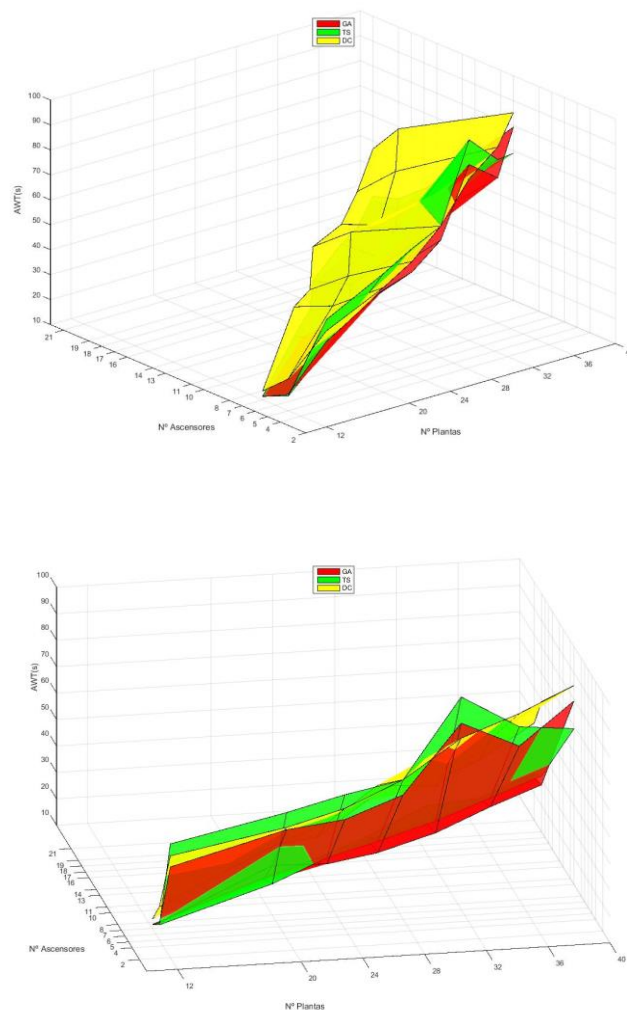


Figura 163: Gráfica 3D N° Ascensores-N° Plantas-AWT (función de evaluación: distancia). Interfloor

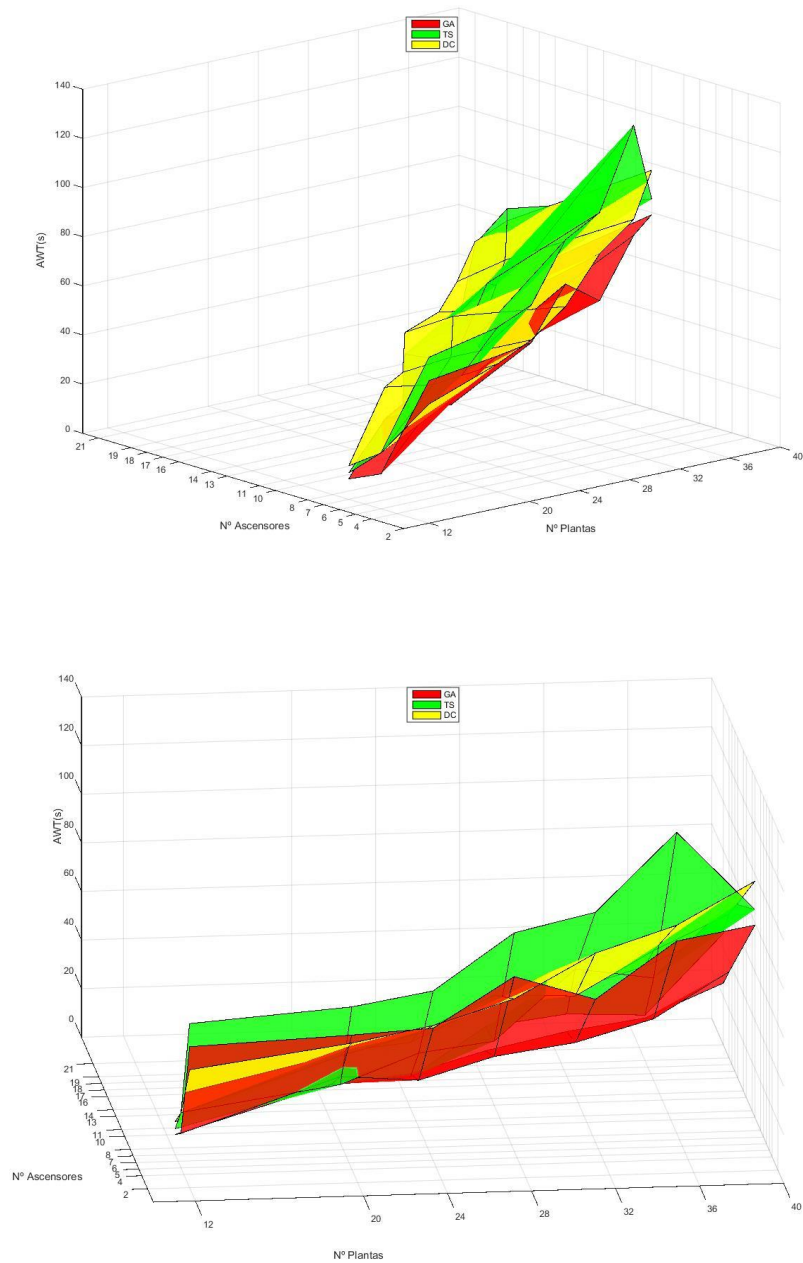


Figura 164: Gráfica 3D N° Ascensores-N° Plantas-AWT (función de evaluación: tiempo). Interfloor

Para el tráfico Interfloor se observa en las Figura 165 y Figura 166 que el mejor tiempo medio de tránsito se obtiene mediante el algoritmo DC-Elevate (color amarillo) tanto para la función de evaluación de distancia como para la basada en tiempo.

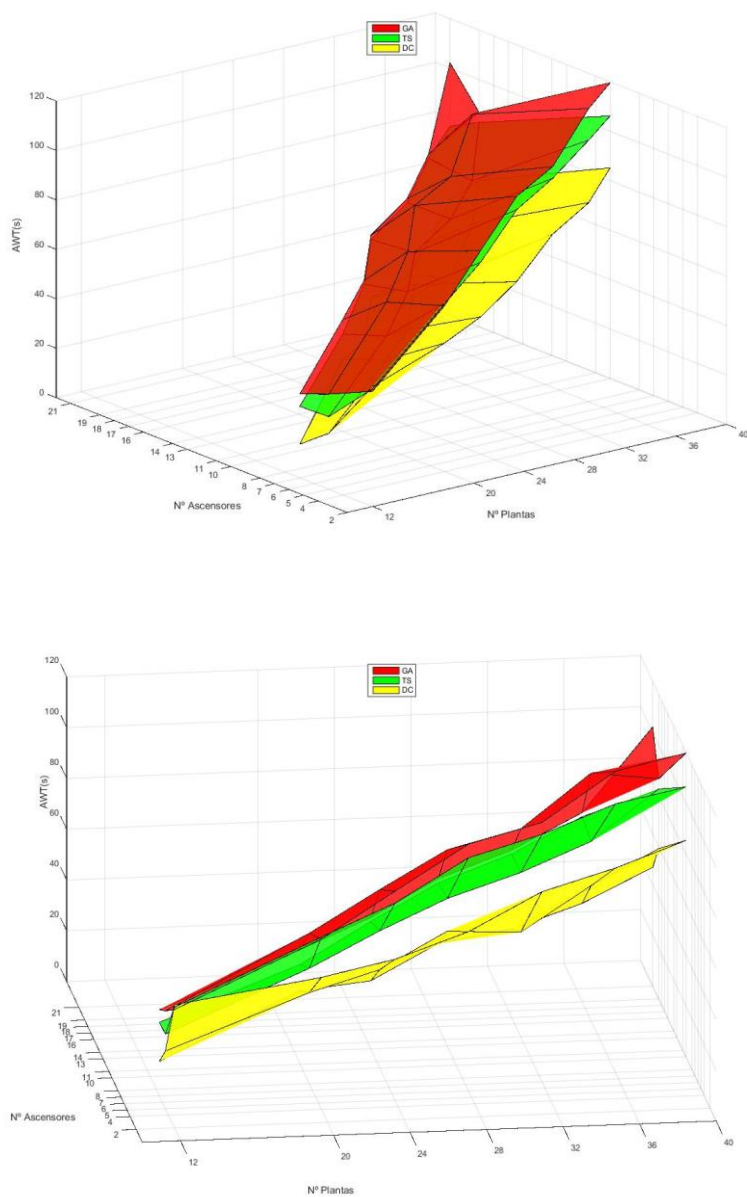


Figura 165: Gráfica 3D Nº Ascensores-Nº Plantas-AJT (función de evaluación: distancia). Interfloor

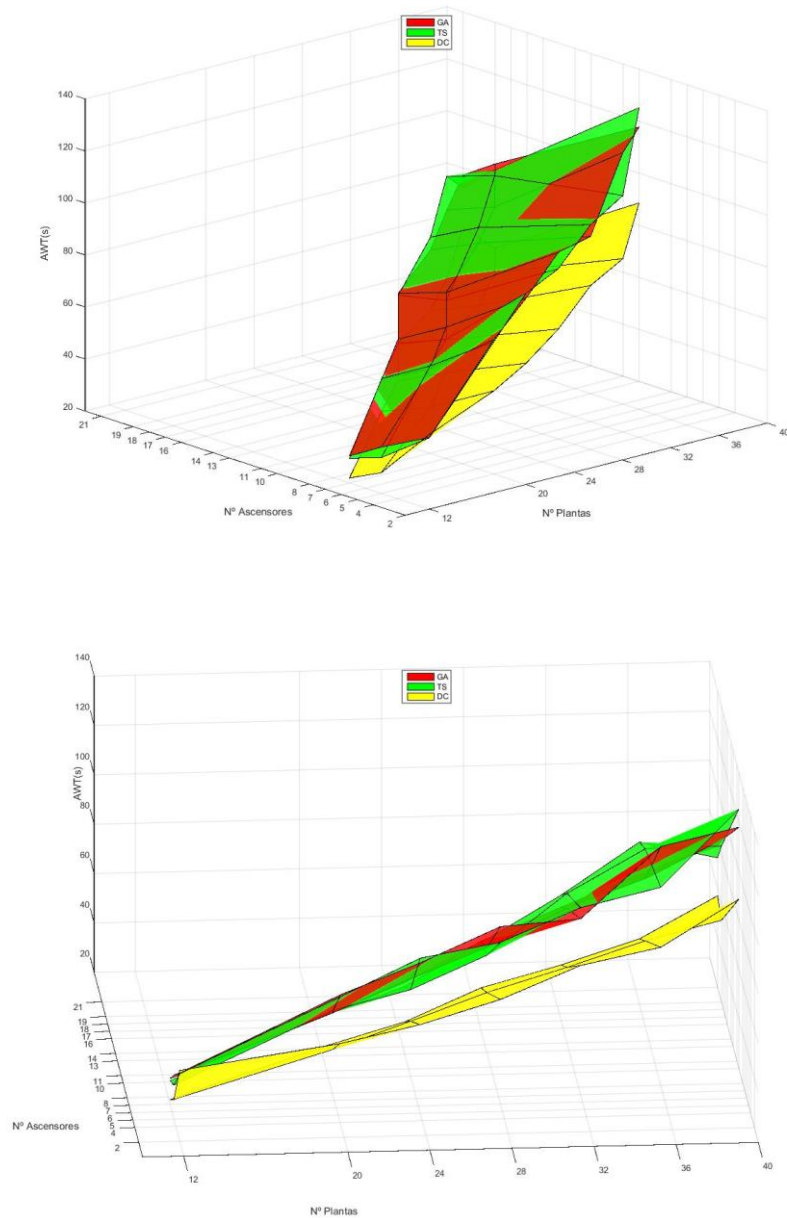


Figura 166: Gráfica 3D Nº Ascensores-Nº Plantas-AJT (función de evaluación: tiempo). Interfloor

8.2. Tráfico Lunchpeak

Para un tipo de tráfico Lunchpeak se observa en la Figura 167, relativa a la función de evaluación basada en distancia, que el tiempo medio de espera se minimiza mediante la implementación del algoritmo genético (color rojo) mientras que el algoritmo de búsqueda tabú (color verde) compite con DC-Elevate (color amarillo) por la segunda posición de mejores tiempos. Para la función de evaluación basada en tiempo se da la misma casuística tal y como se observa en la Figura 168.

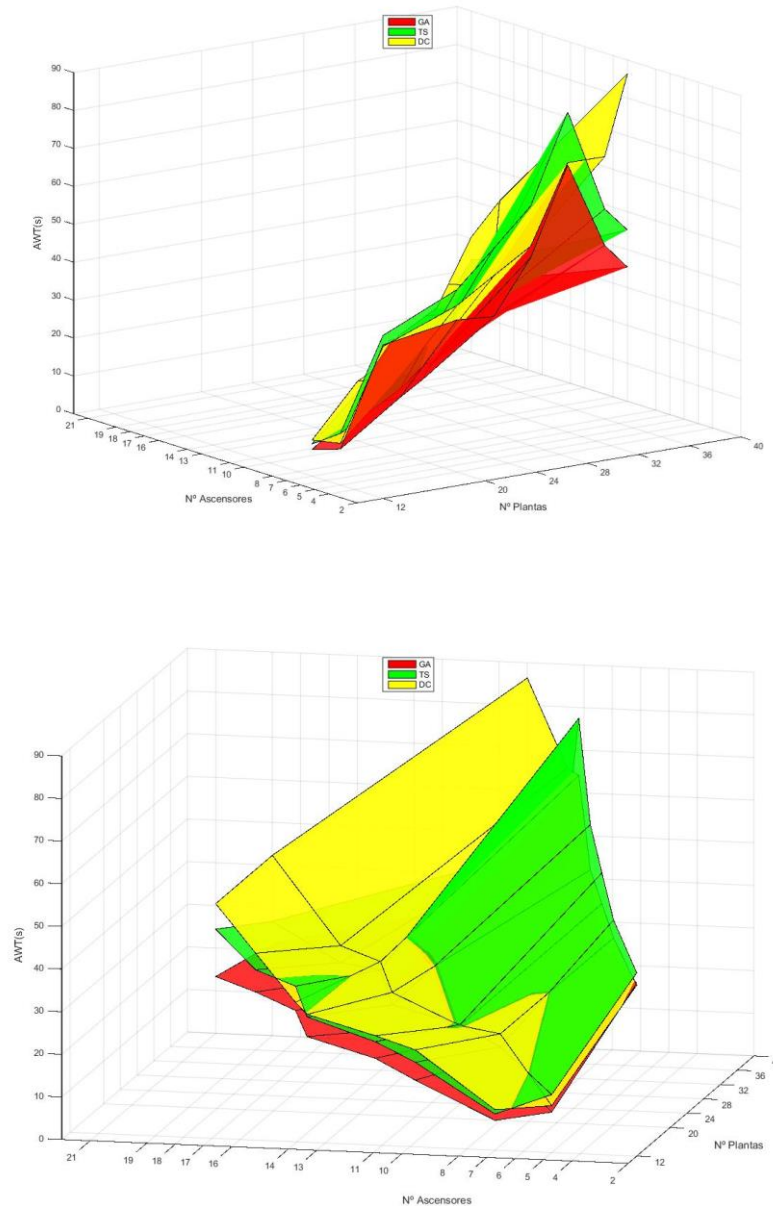


Figura 167: Gráfica 3D Nº Ascensores-Nº Plantas-AWT (función de evaluación: distancia). Lunchpeak

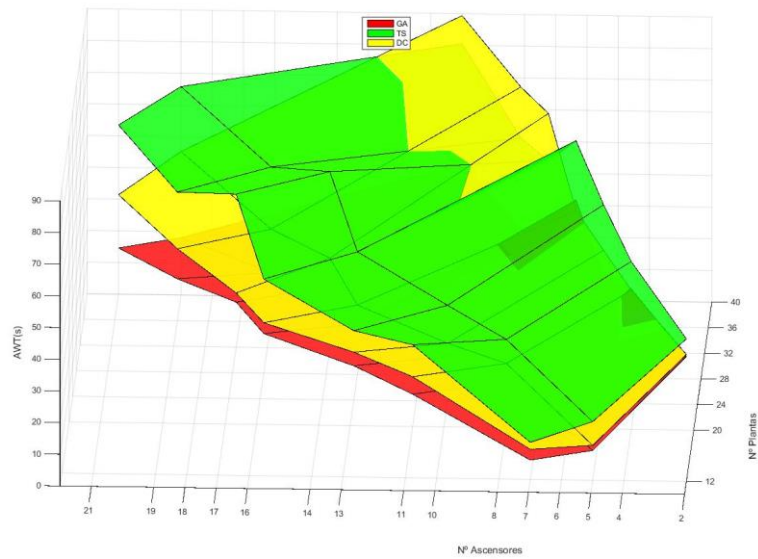
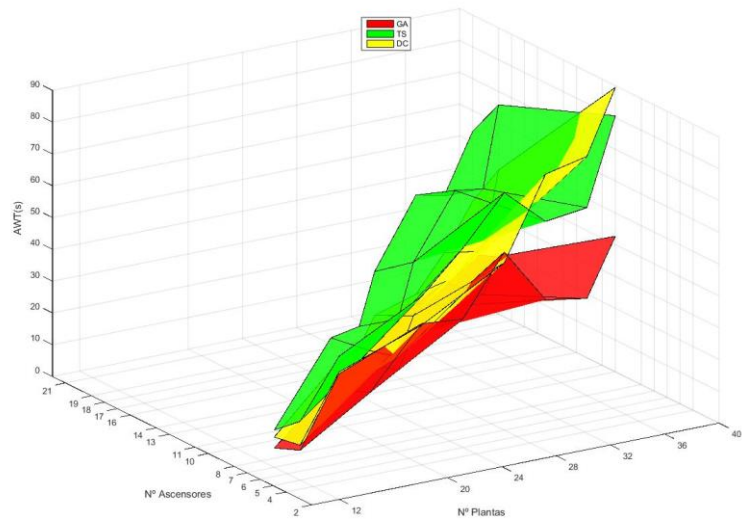


Figura 168: Gráfica 3D N° Ascensores-N° Plantas-AWT (función de evaluación: tiempo). Lunchpeak

Para el tiempo medio de tránsito se observa en la Figura 169 y Figura 170 que con el patrón de tráfico Lunchpeak se obtienen los mejores resultados mediante la implementación del algoritmo DC-Elevate (color amarillo) tanto para la función de evaluación basada en distancia como tiempo.

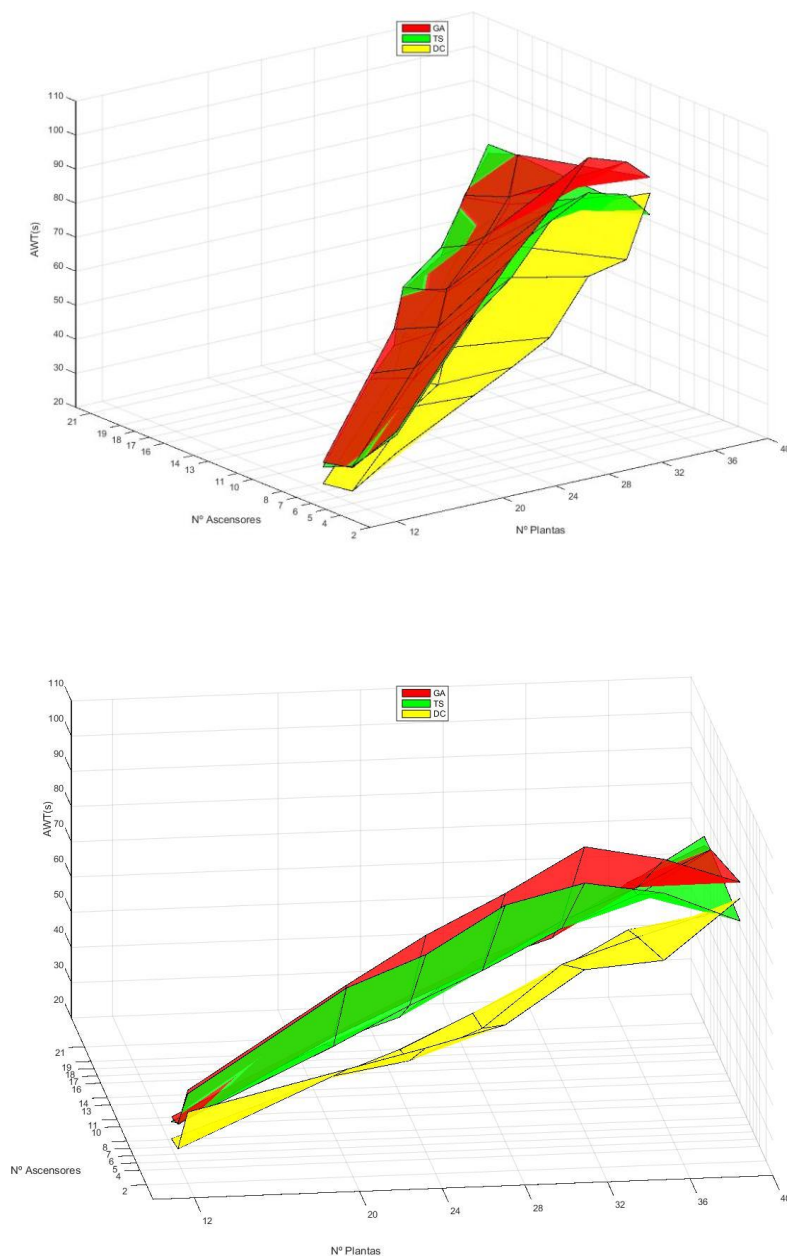


Figura 169: Gráfica 3D Nº Ascensores-Nº Plantas-AJT (función de evaluación: distancia). Lunchpeak

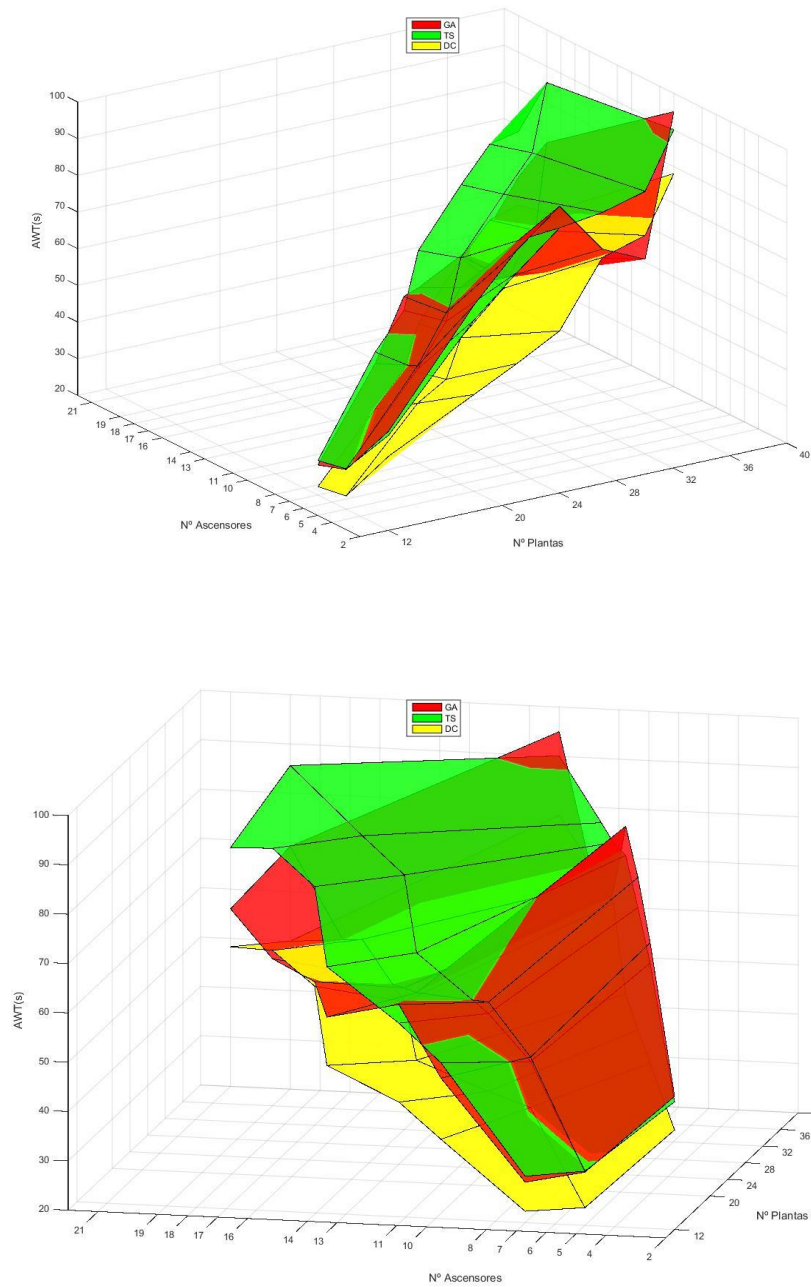


Figura 170: Gráfica 3D N° Ascensores-N° Plantas-AJT (función de evaluación: tiempo). Lunchpeak

8.3. Tráfico Downpeak

Para el tipo de tráfico Downpeak se muestra en la Figura 171 que el tiempo medio de espera de pasajeros es minimizado mediante el algoritmo genético (color rojo) seguido del algoritmo de búsqueda tabú (color verde). Sin embargo, para la función de evaluación basada en tiempo se obtiene que la minimización la consigue el algoritmo genético seguido de DC-Elevate (color amarillo), aunque este último genera un pico elevado de tiempo para la casuística de 40 plantas, como se aprecia en la Figura 172.

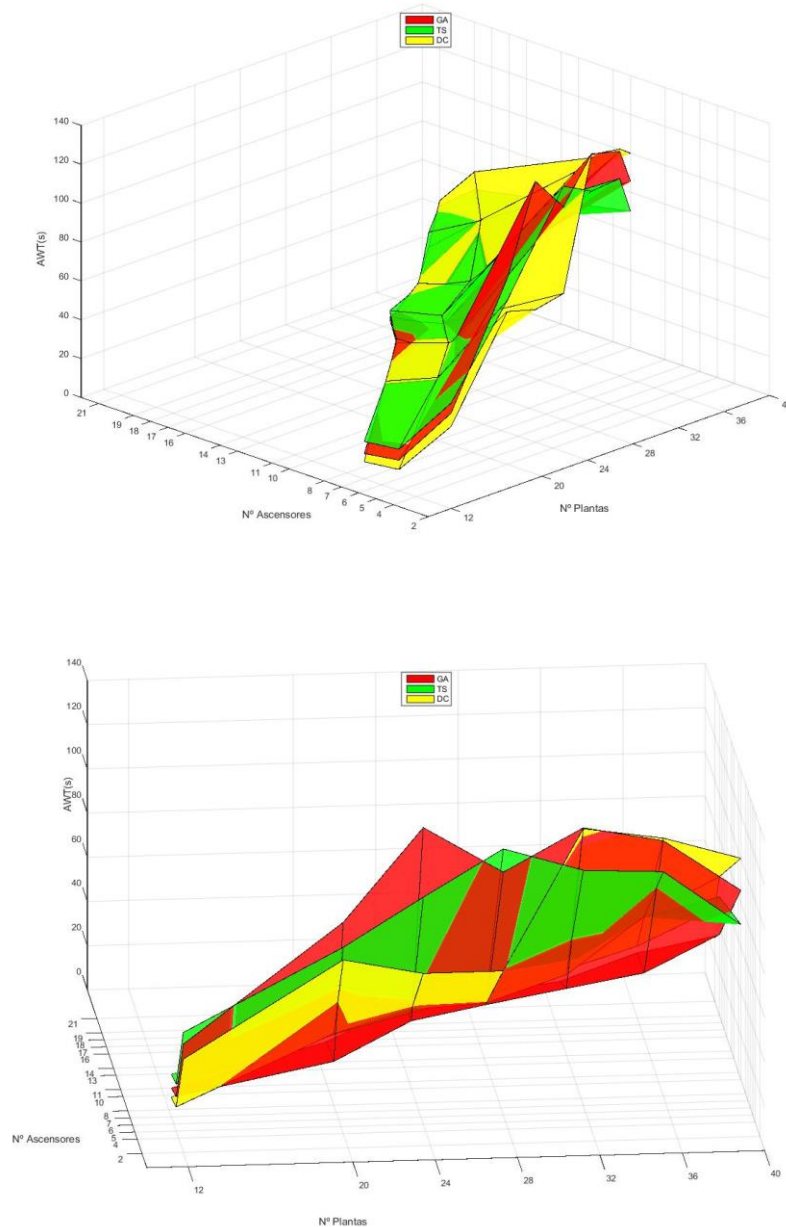


Figura 171: Gráfica 3D N° Ascensores-N° Plantas-AWT (función de evaluación: distancia). Downpeak

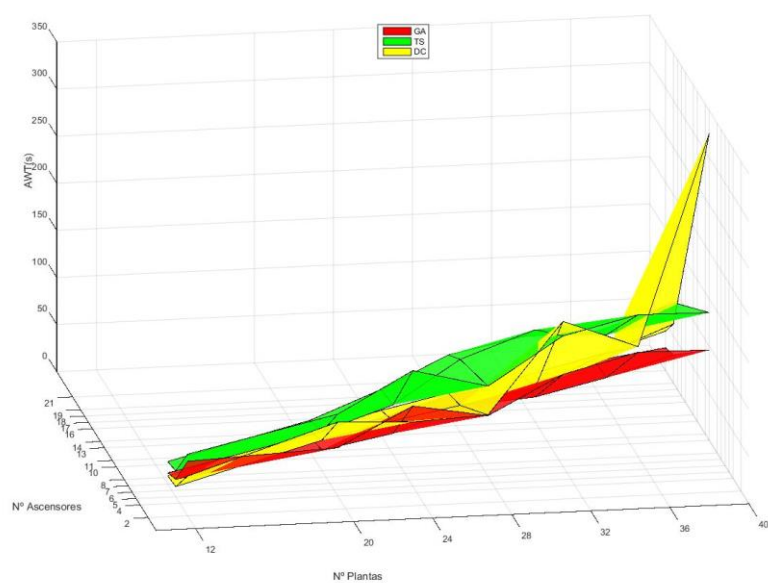
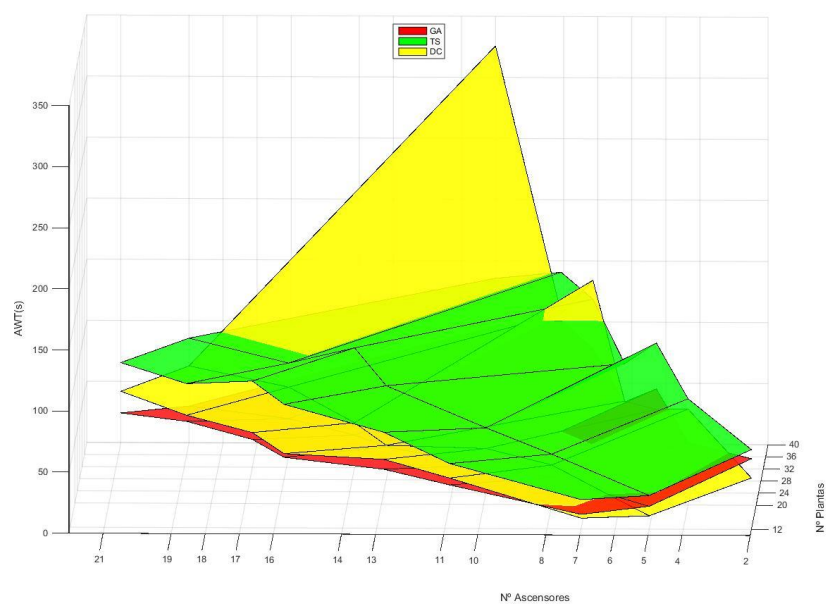


Figura 172: Gráfica 3D N° Ascensores-N° Plantas-AWT (función de evaluación: tiempo). Downpeak

Con respecto al tiempo medio de tránsito para el tipo de tráfico Downpeak, se aprecia en las Figura 173 y Figura 174 que aunque el algoritmo implementado por *Elevate* (color amarillo) aporta los mejores resultados, estos no se alejan demasiado de los tiempos que se obtienen mediante la búsqueda tabú (color verde) y el algoritmo genético (color rojo).

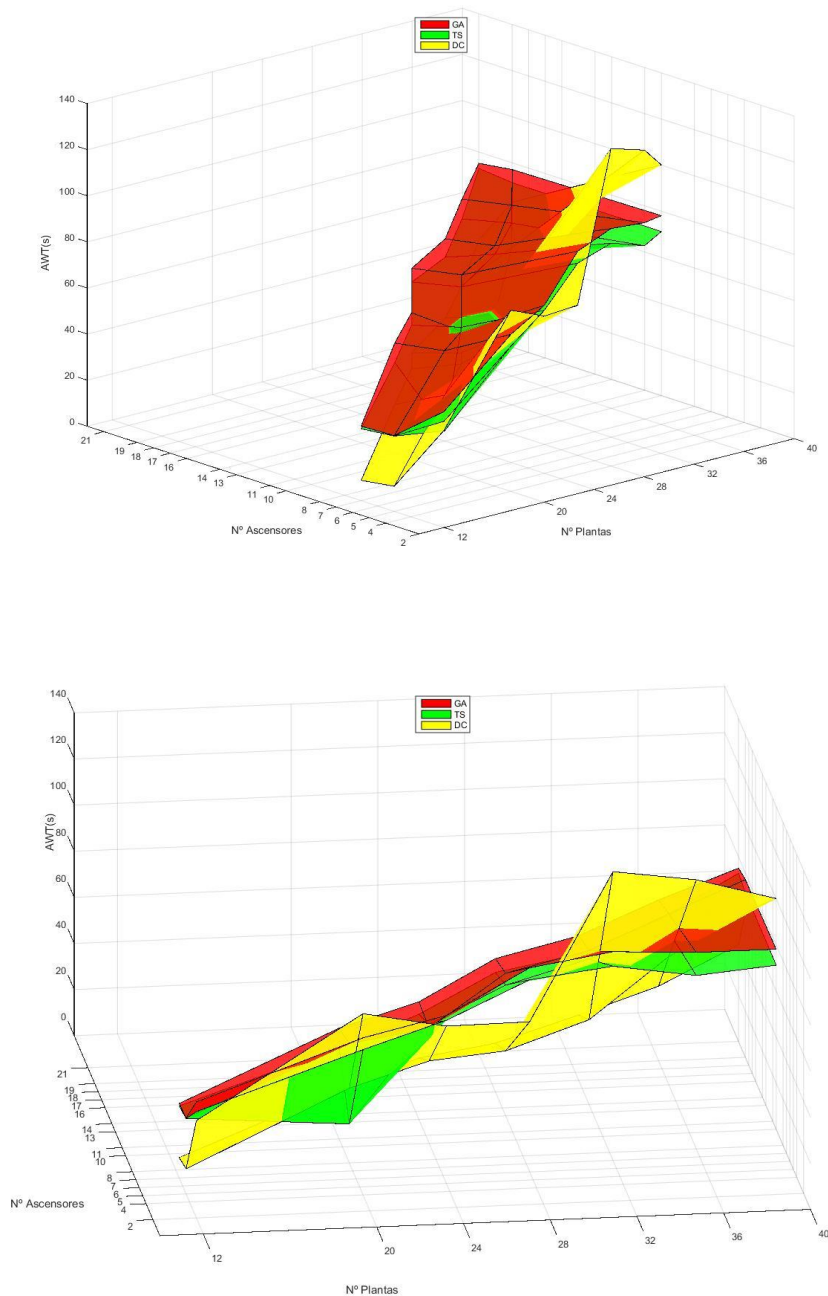


Figura 173: Gráfica 3D Nº Ascensores-Nº Plantas-AJT (función de evaluación: distancia). Downpeak

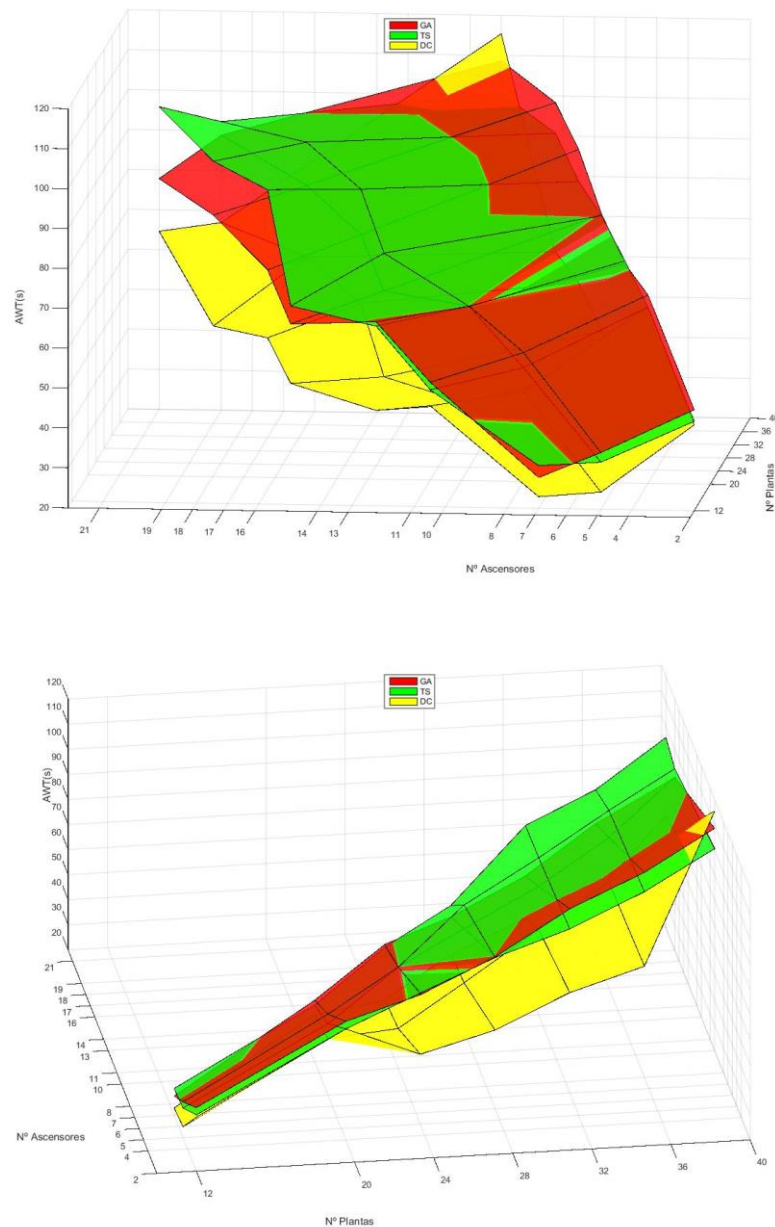



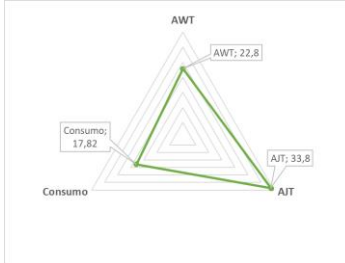

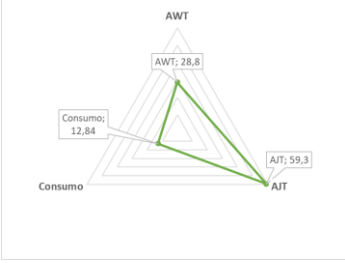
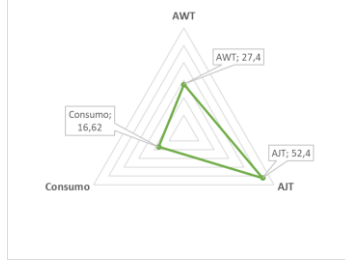

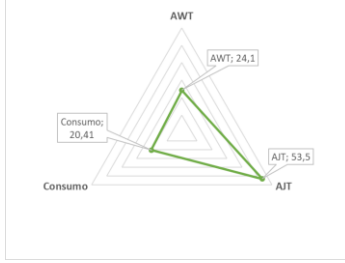
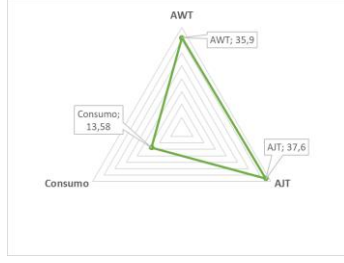
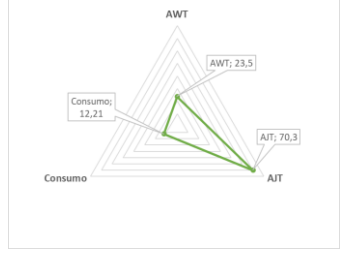
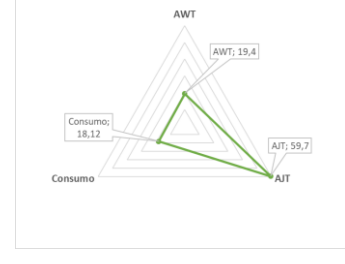
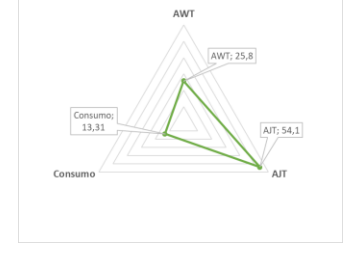






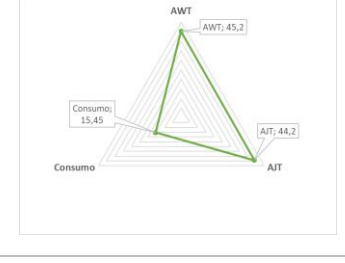


Figura 174: Gráfica 3D N° Ascensores-N° Plantas-AJT (función de evaluación: tiempo). Downpeak

8.4. Propuesta de aplicación de parámetros de desempeño

Se recoge en este apartado el estudio de los parámetros de desempeño de los sistemas de ascensores de modo que se pueda definir para cada combinación de número de plantas y número de ascensores, la función de evaluación más favorable, así como el método de resolución. Las siguientes tablas muestran las gráficas radiales obtenidas tras combinar el tiempo medio de espera (AWT), el tiempo medio de tránsito (AJT) y el consumo energético (Consumo).

El área que encierra cada una de las gráficas sirve para poder comparar diferentes combinaciones, de modo que, si una combinación tiene un área mayor que otra combinación, se puede afirmar que la de menor área muestra un mejor desempeño. Se demuestra que el desempeño es mejor debido a que si el área es menor, esto quiere decir que al menos una de las variables (AWT, AJT, Consumo) que forman dicha área ha obtenido un valor menor.

Tabla 19: Propuesta de aplicación Interfloor	GA-Distancia	GA-Tiempo	TS-Distancia	TS-Tiempo	DC-Elevate
12 Plantas 5 Cabinas					
20 Plantas 8 Cabinas					
24 Plantas 10 Cabinas					
28 Plantas 13 Cabinas					

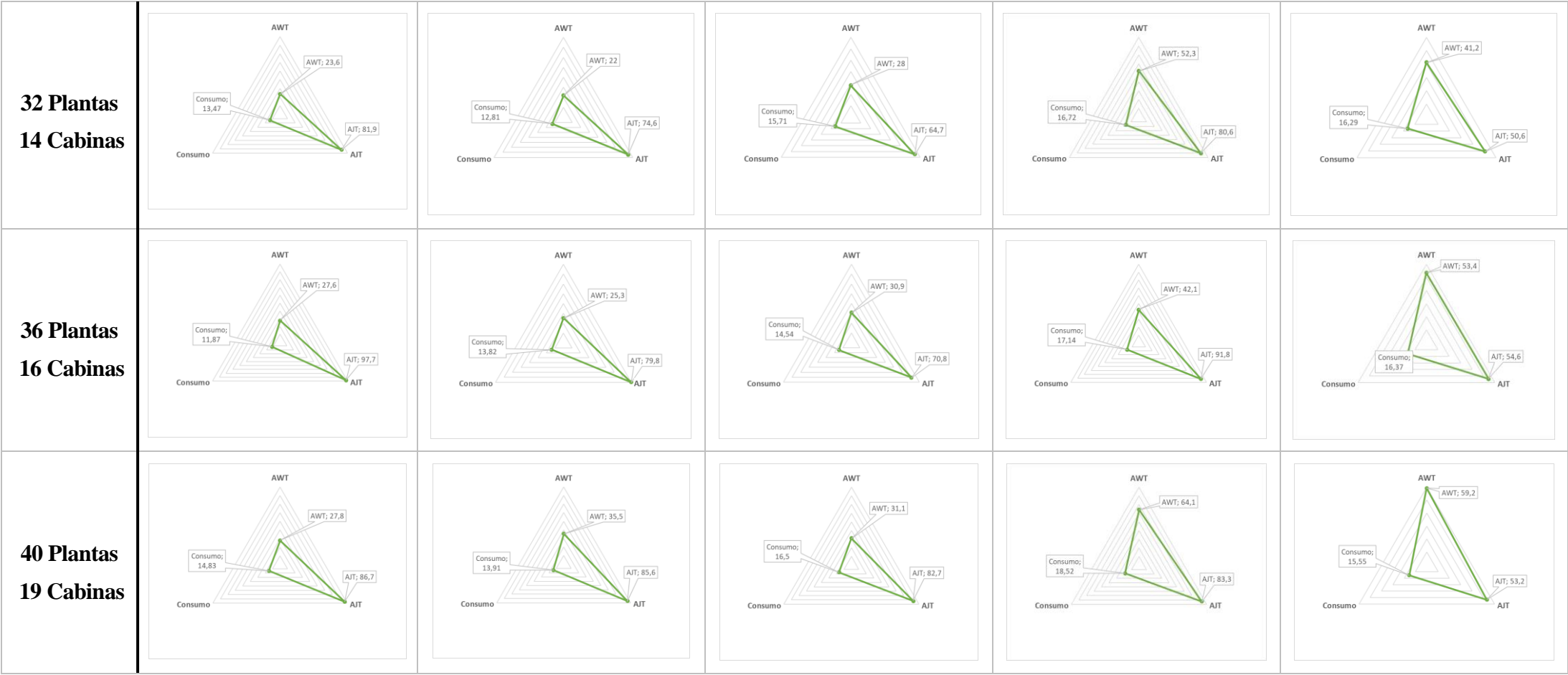



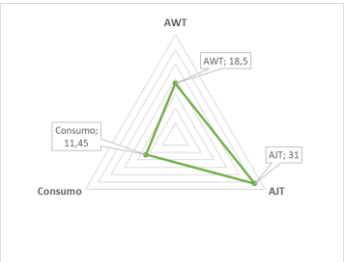
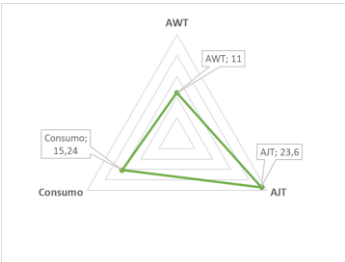


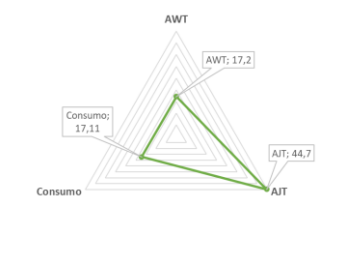





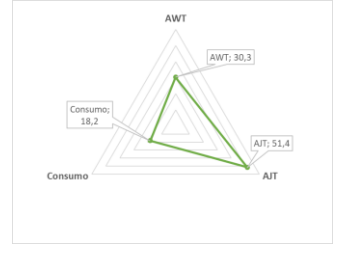








Tabla 20: Propuesta de aplicación Lunchpeak	GA-Distancia	GA-Tiempo	TS-Distancia	TS-Tiempo	DC-Elevate
12 Plantas 5 Cabinas					
20 Plantas 8 Cabinas					
24 Plantas 10 Cabinas					
28 Plantas 13 Cabinas					

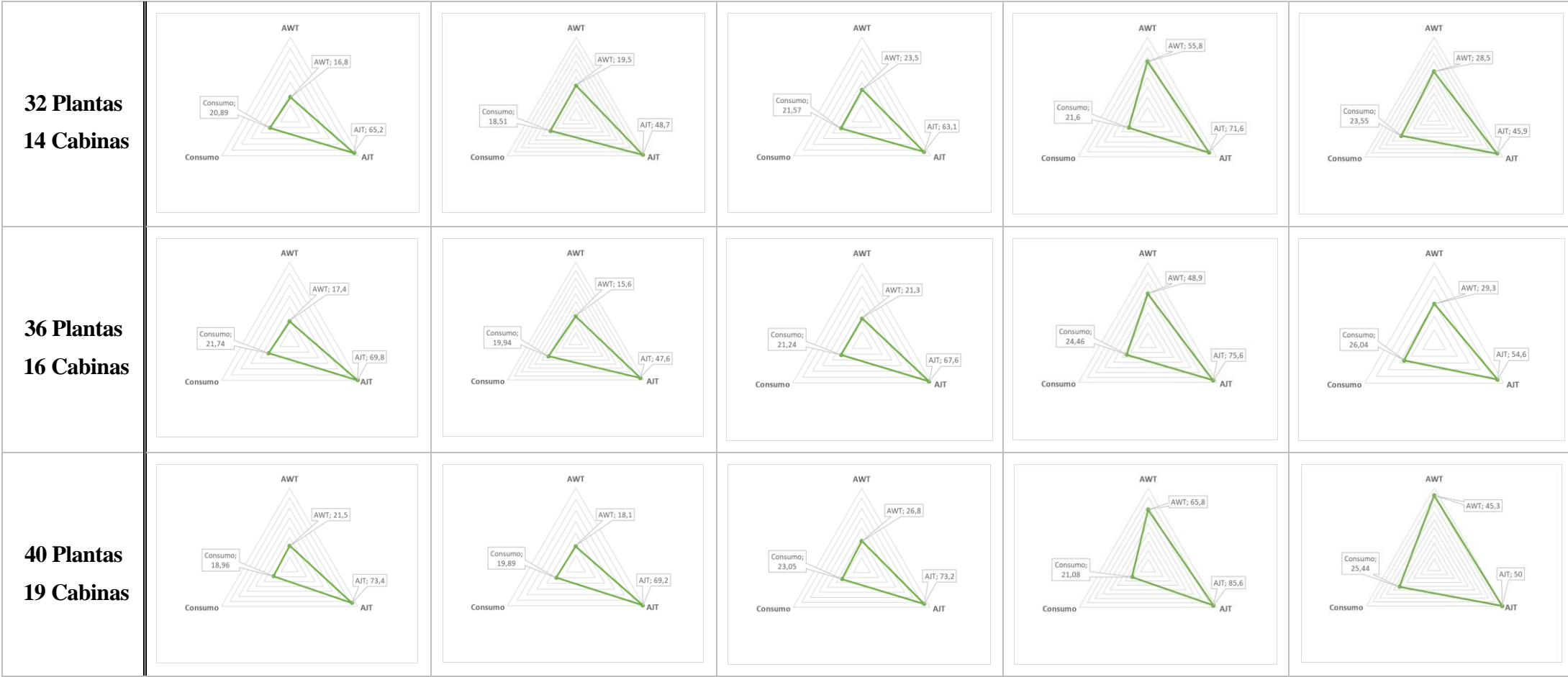
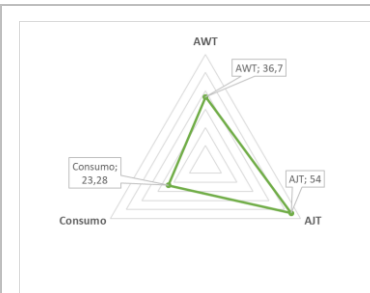
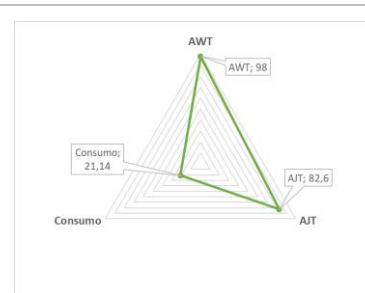
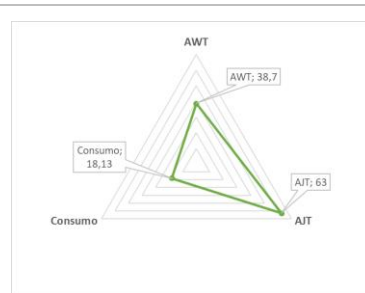
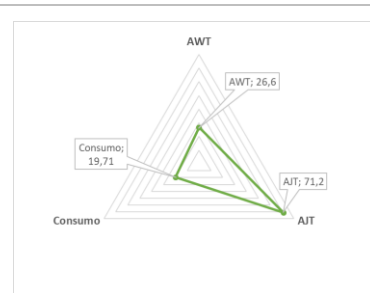
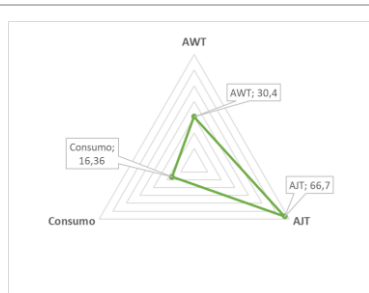
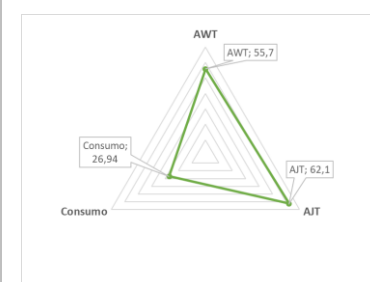
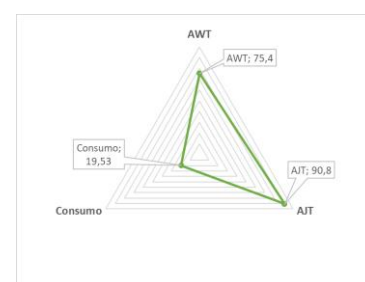
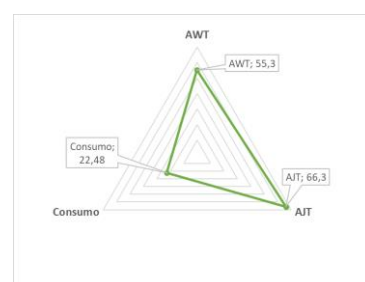
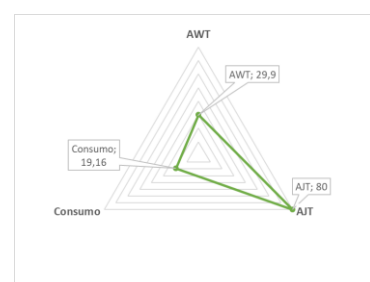
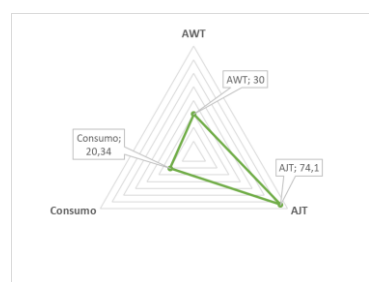


Tabla 21: Propuesta de aplicación Downpeak	GA-Distancia	GA-Tiempo	TS-Distancia	TS-Tiempo	DC-Elevate
12 Plantas 5 Cabinas					
20 Plantas 8 Cabinas					
24 Plantas 10 Cabinas					
28 Plantas 13 Cabinas					

**32 Plantas
14 Cabinas**



**36 Plantas
16 Cabinas**



**40 Plantas
19 Cabinas**

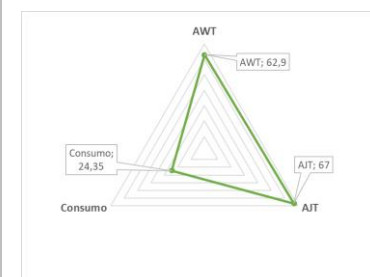
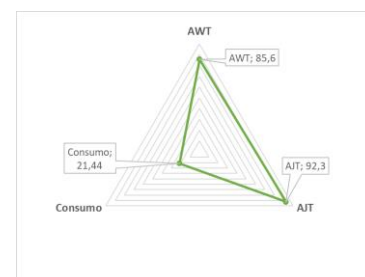
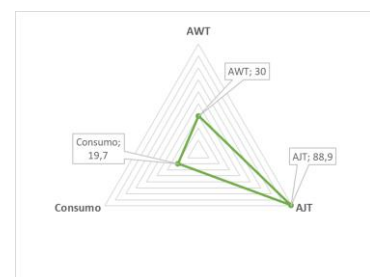
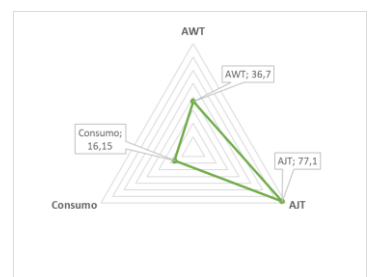
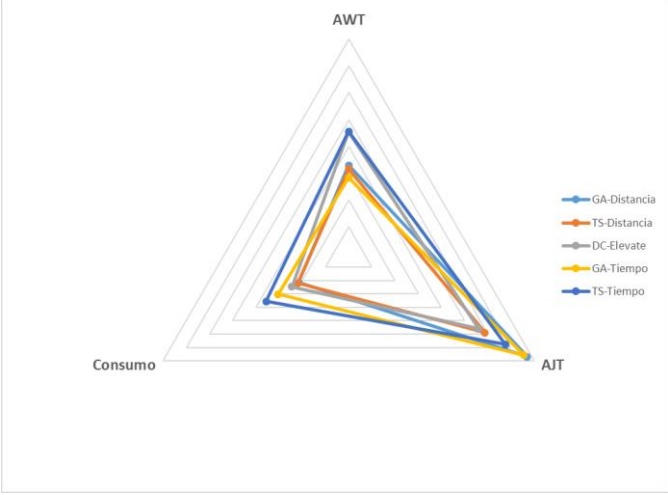
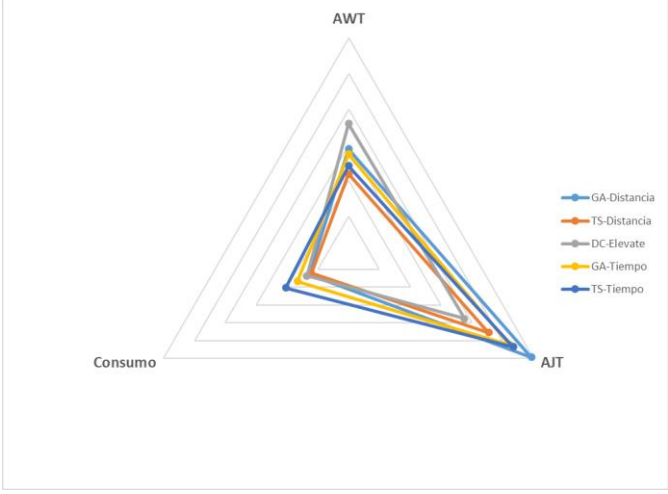
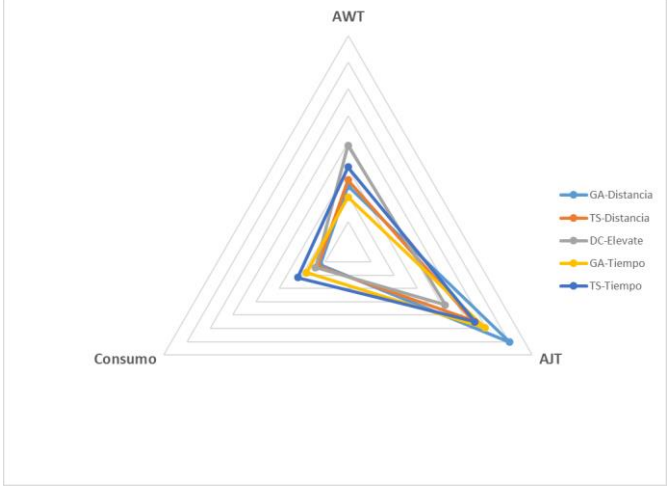
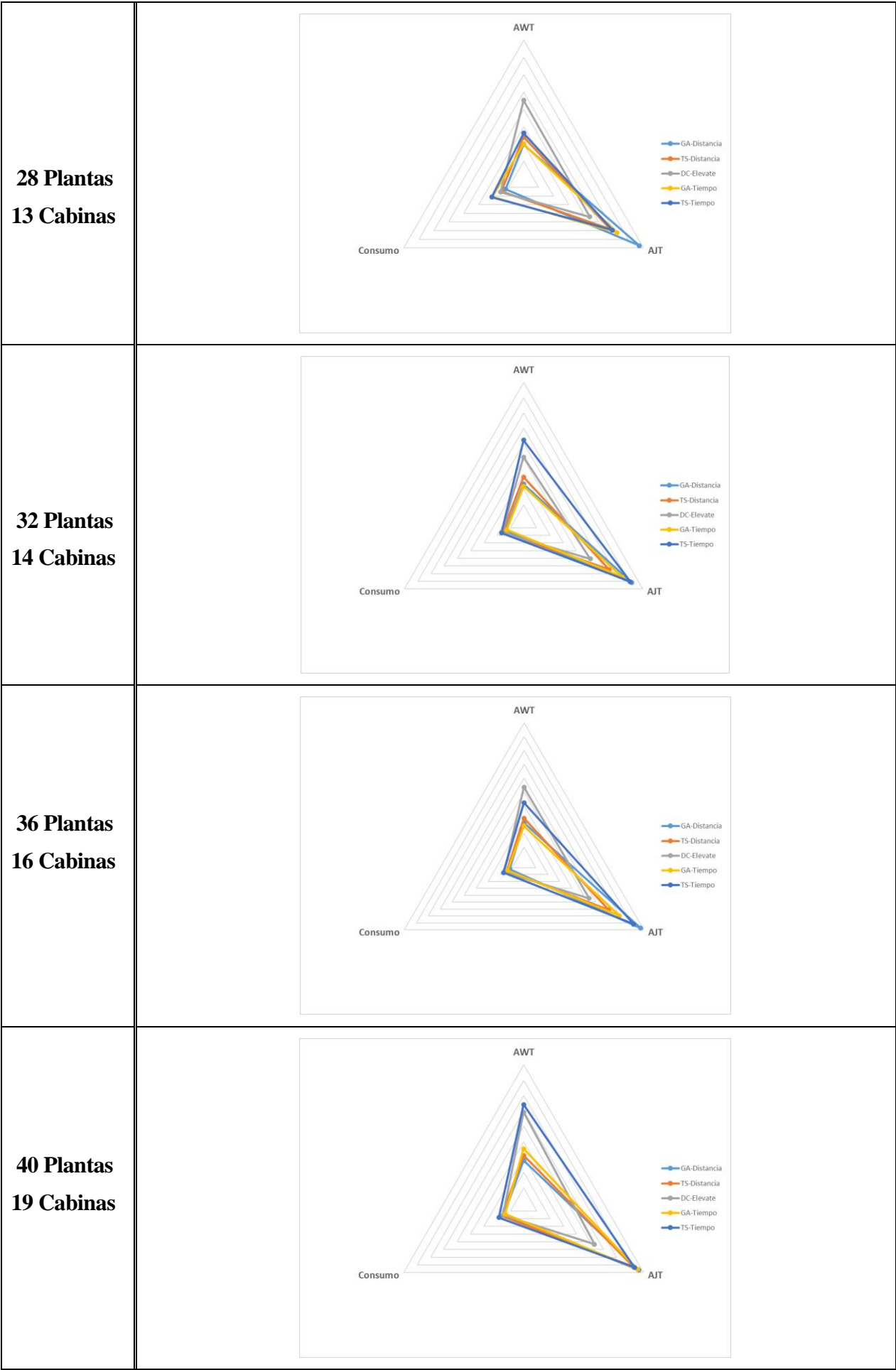
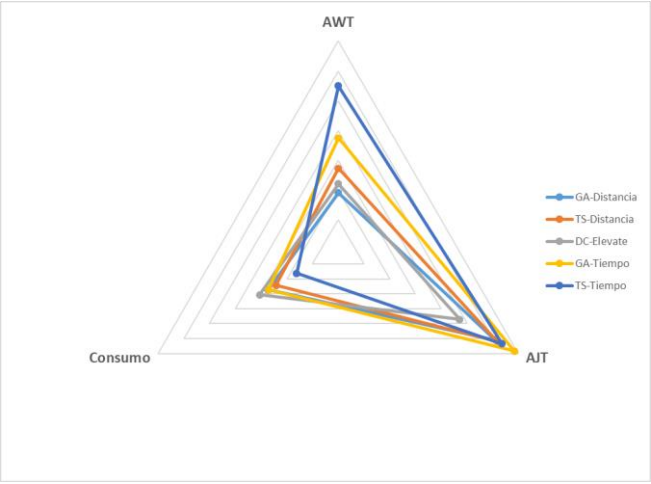
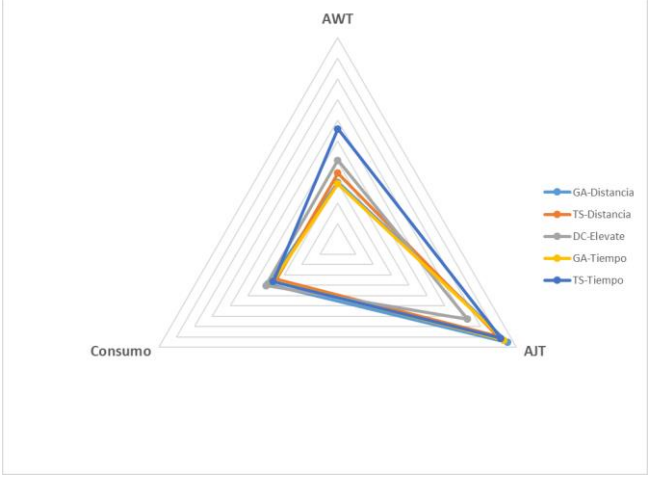
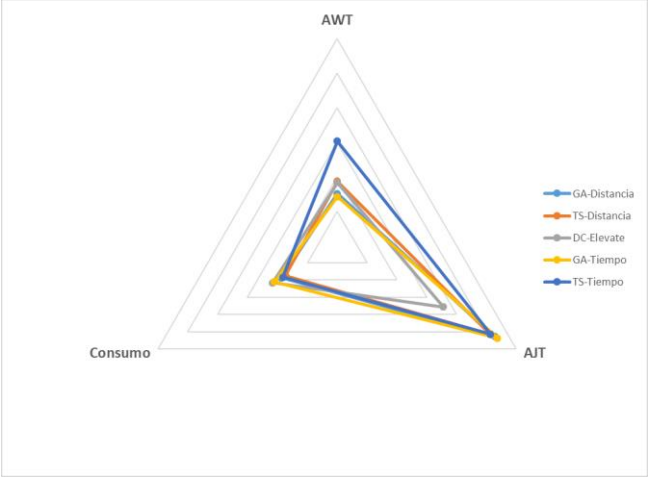
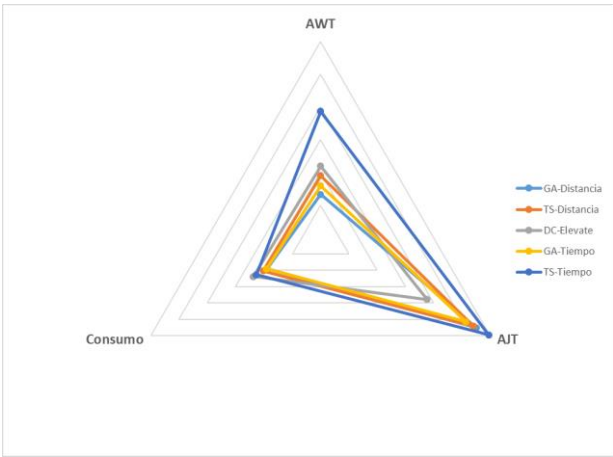


Tabla 22: Comparativa resolución- evaluación para Interfloor	Métodos de resolución y funciones de evaluación
12 Plantas 5 Cabinas	 <p>A radar chart comparing five methods (GA-Distancia, TS-Distancia, DC-Elevate, GA-Tiempo, TS-Tiempo) across three metrics: AWT (top), Consumo (bottom-left), and AJT (bottom-right). The chart shows concentric lines for each method, with GA-Tiempo and TS-Tiempo generally showing the highest values across all metrics.</p>
20 Plantas 8 Cabinas	 <p>A radar chart comparing five methods (GA-Distancia, TS-Distancia, DC-Elevate, GA-Tiempo, TS-Tiempo) across three metrics: AWT (top), Consumo (bottom-left), and AJT (bottom-right). The chart shows concentric lines for each method, with GA-Tiempo and TS-Tiempo generally showing the highest values across all metrics.</p>
24 Plantas 10 Cabinas	 <p>A radar chart comparing five methods (GA-Distancia, TS-Distancia, DC-Elevate, GA-Tiempo, TS-Tiempo) across three metrics: AWT (top), Consumo (bottom-left), and AJT (bottom-right). The chart shows concentric lines for each method, with GA-Tiempo and TS-Tiempo generally showing the highest values across all metrics.</p>

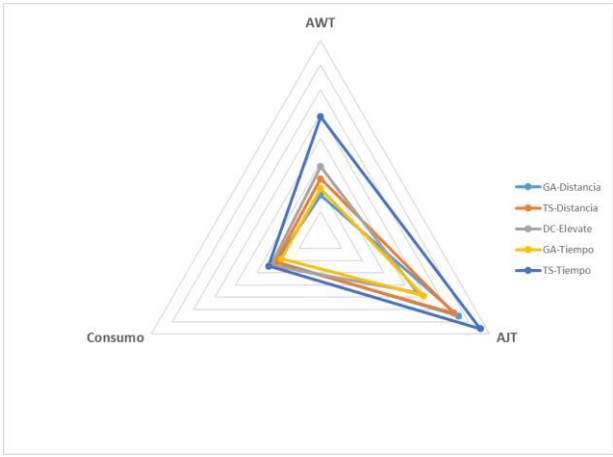


<p>Tabla 23: Comparativa resolución- evaluación para Lunchpeak</p>	<p>Métodos de resolución y funciones de evaluación</p>
<p>12 Plantas 5 Cabinas</p>	 <p>A radar chart comparing five methods (GA-Distancia, TS-Distancia, DC-Elevate, GA-Tiempo, TS-Tiempo) across three axes: AWT (top), Consumo (bottom-left), and AJT (bottom-right). The chart shows concentric rings representing different levels of performance. GA-Tiempo and TS-Tiempo generally show the highest performance (outermost rings) across all three axes, while DC-Elevate shows the lowest performance (innermost rings).</p>
<p>20 Plantas 8 Cabinas</p>	 <p>A radar chart comparing five methods (GA-Distancia, TS-Distancia, DC-Elevate, GA-Tiempo, TS-Tiempo) across three axes: AWT (top), Consumo (bottom-left), and AJT (bottom-right). The chart shows concentric rings representing different levels of performance. GA-Tiempo and TS-Tiempo generally show the highest performance (outermost rings) across all three axes, while DC-Elevate shows the lowest performance (innermost rings).</p>
<p>24 Plantas 10 Cabinas</p>	 <p>A radar chart comparing five methods (GA-Distancia, TS-Distancia, DC-Elevate, GA-Tiempo, TS-Tiempo) across three axes: AWT (top), Consumo (bottom-left), and AJT (bottom-right). The chart shows concentric rings representing different levels of performance. GA-Tiempo and TS-Tiempo generally show the highest performance (outermost rings) across all three axes, while DC-Elevate shows the lowest performance (innermost rings).</p>

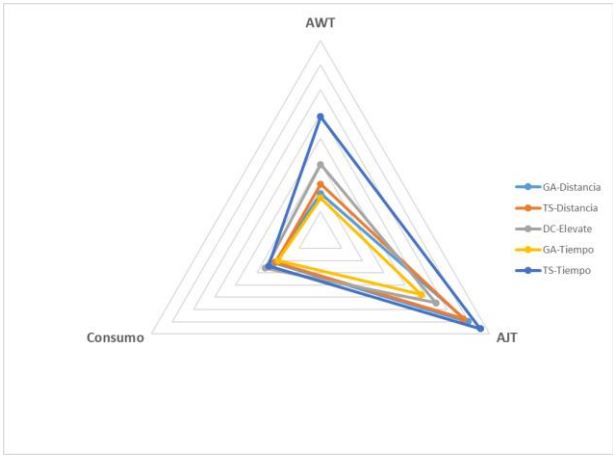
**28 Plantas
13 Cabinas**



**32 Plantas
14 Cabinas**



**36 Plantas
16 Cabinas**



**40 Plantas
19 Cabinas**

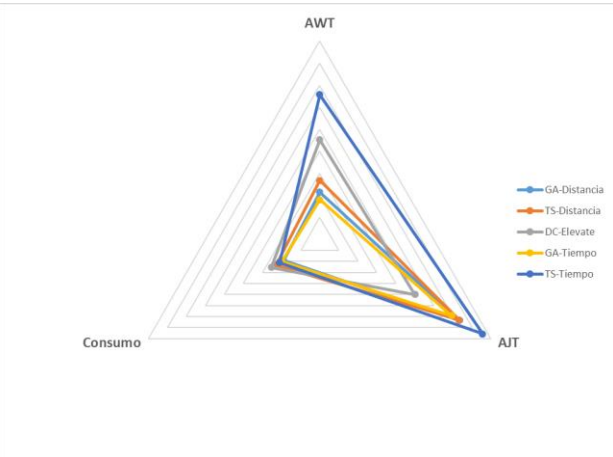
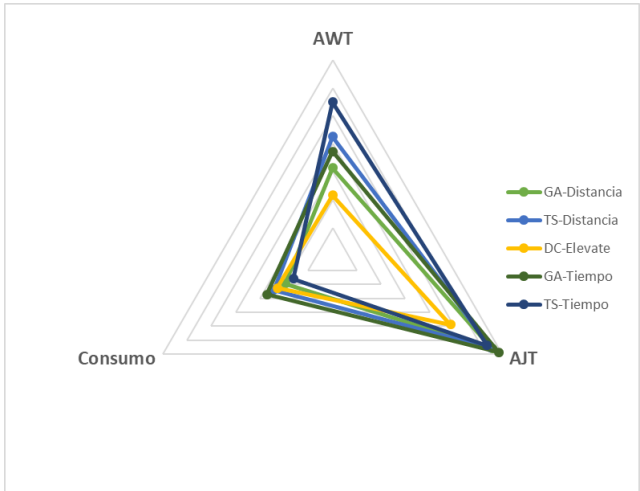


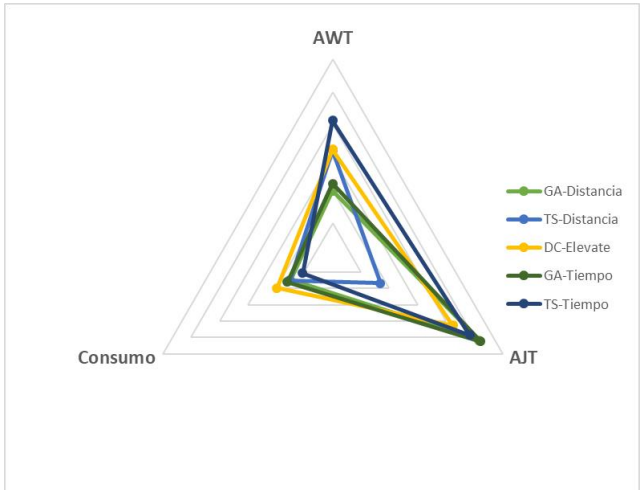
Tabla 24:
Comparativa
resolución-
evaluación
para
Downpeak

Métodos de evaluación y funciones de evaluación

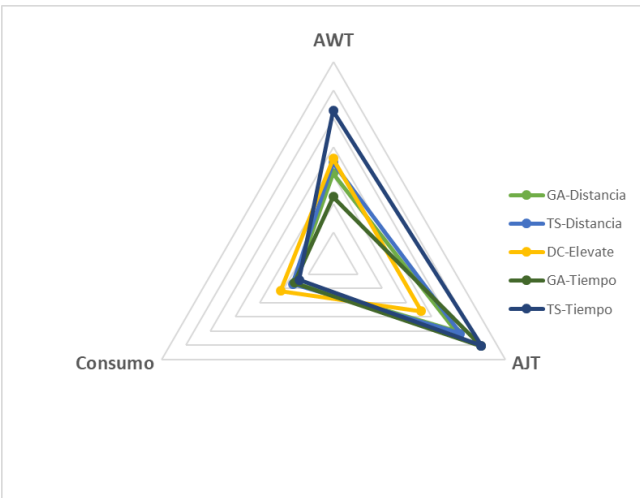
**12 Plantas
5 Cabinas**



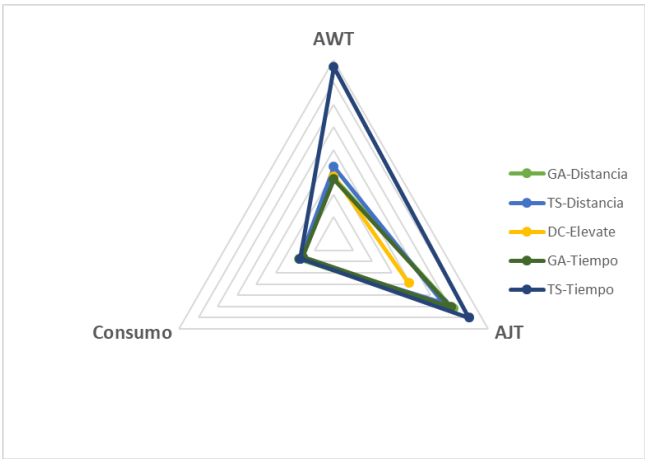
**20 Plantas
8 Cabinas**



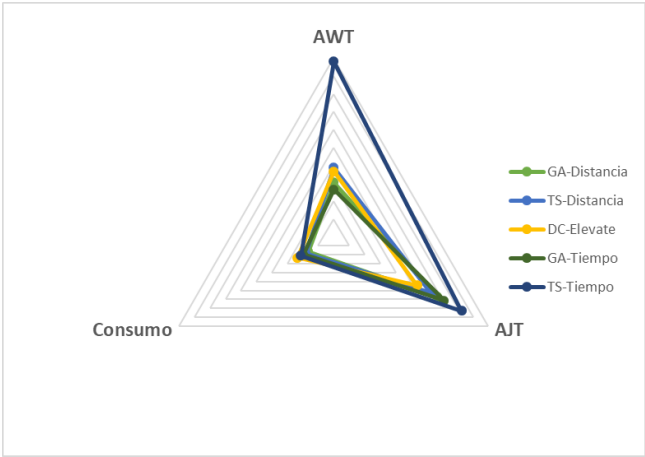
**24 Plantas
10 Cabinas**



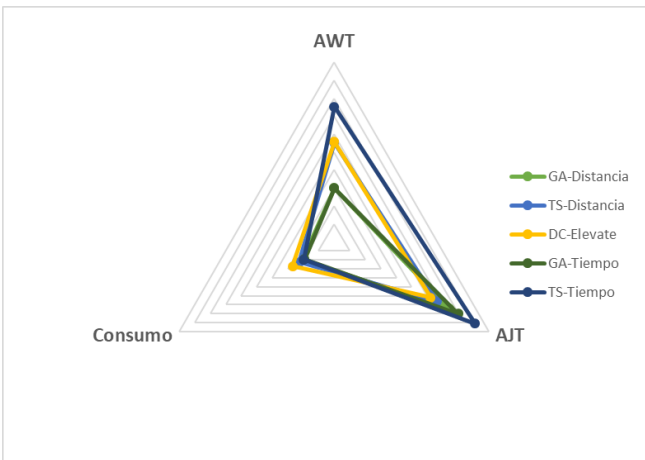
**28 Plantas
13 Cabinas**



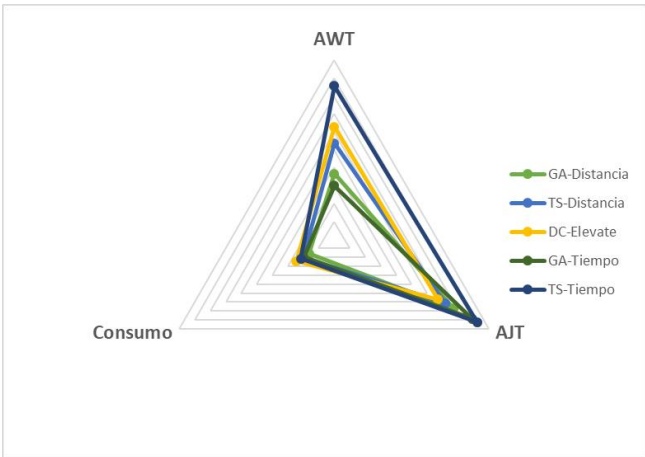
**32 Plantas
14 Cabinas**



**36 Plantas
16 Cabinas**



**40 Plantas
19 Cabinas**



9. CONCLUSIONES

El incremento del precio del metro cuadrado de terreno hace que el tráfico vertical adquiriera una gran relevancia en la actualidad debido a que el aprovechamiento del espacio es una necesidad. De la necesidad de aprovechamiento del espacio surge el incremento del número de plantas de los edificios de modo que se hace esencial la importancia del transporte, tanto de personas como de mercancías, a través del edificio.

El transporte vertical se consigue mediante ascensores, pero debido al incremento del número de plantas de los edificios, la tecnología que se emplea para dicho transporte se encuentra en constante desarrollo para lograr la disminución del tiempo de espera y tránsito de los pasajeros, así como para conseguir un modelo de transporte seguro.

La evolución del transporte vertical ha logrado una mejora tanto en la forma de transporte, más silenciosa y rápida, como en la estética gracias a la integración del mismo como un recurso arquitectónico de decoración. Además, en consecuencia, del estudio de los patrones de tráfico se tiene que se ha mejorado tras la identificación de los cuatro patrones esenciales (interfloor, uppeak, downpeak y lunchpeak). Es importante reseñar que el estudio del patrón uppeak no se lleva a cabo en este trabajo debido a que es un tipo de patrón de tráfico peculiar por tener que enviar en todo momento las cabinas hacia la planta baja del edificio, encontrándose las metaheurísticas programadas optimizadas para los otros tres tipos de tráfico. Además, existen diversidad de estudios con respecto al patrón de tráfico uppeak mientras que para los otros tres tipos de patrones se tiene una menor cantidad de trabajos publicados.

El ascensor con arquitectura *Double Deck* nace del aumento del flujo de pasajeros en grandes edificios, pero requiere de un sistema de asignación de cabinas a llamadas eficiente. Se requiere un algoritmo de asignación de llamadas que sirva todas las llamadas del edificio logrando que el tiempo de espera de los pasajeros frente al ascensor y el tiempo que estos tardan en llegar a su destino sea el menor posible.

Se muestra en la Tabla 22, Tabla 23 y Tabla 24, para cada combinación de número de plantas y número de ascensores de cada patrón de tráfico, el conjunto de áreas combinando cada una de las cinco formas de resolución-evaluación (algoritmo genético con función de evaluación de distancia, algoritmo genético con función de evaluación de tiempo, búsqueda tabú con función de evaluación de distancia, búsqueda tabú con función de evaluación de tiempo y DC-Elevate). Se aprecia de este modo para los tres factores clave (tiempo medio de espera, tiempo medio de tránsito y consumo energético) cuales son los mejores parámetros de desempeño.

Para un tipo de tráfico interfloor, se observa cómo las áreas de la búsqueda tabú con función de evaluación de tiempo y DC-Elevate son, por lo general, mayores que las del resto. En cambio, el algoritmo genético con función de evaluación basada en tiempo y la búsqueda tabú con función de evaluación de distancia obtienen las menores áreas para este tipo de tráfico.

Para un tipo de tráfico lunchpeak, se observa que las mayores áreas son obtenidas por la búsqueda tabú con función de evaluación de tiempo. Los mejores resultados son obtenidos mediante el algoritmo genético en sus versiones de función de evaluación basada en distancia y basada en tiempo.

Para un tipo de tráfico downpeak, las peores combinaciones, es decir, las que han obtenido unas mayores áreas son las formadas por la búsqueda tabú basada en tiempo y el algoritmo implementado por *Elevate*. Sin embargo, el algoritmo genético con función de evaluación basada en tiempo obtiene los mejores resultados.

Es evidenciable mediante el estudio realizado que, para cualquiera de los tres tipos de tráfico analizados, el algoritmo genético aporta los mejores resultados para el tiempo medio de espera de pasajeros frente al ascensor. Se obtienen, además, mejores tiempos si se combina el algoritmo genético con la función de evaluación basada en tiempo en vez de la función de evaluación basada en distancia.

La observación de la Tabla 19, Tabla 20 y Tabla 21 concluye que para cada combinación de plantas/ascensores (observación por filas) el algoritmo genético obtiene las menores áreas encerradas en casi todos los casos por lo que se corrobora la afirmación citada de que es el algoritmo genético la metaheurística que mejor desempeño aporta en el estudio de la asignación de llamadas.

Se obtiene mediante el algoritmo genético un nivel de servicio, para el tiempo medio de espera, catalogado como Bueno/Excelente según el apartado 3.2. ya que gran parte de las llamadas son atendidas en menos de 30 segundos, siendo el número mayor que esta cifra para las combinaciones de edificios con la menor cantidad de cabinas.

Para el tiempo medio de tránsito, es el algoritmo DC-Elevate el que ofrece los mejores resultados. Tras analizar el consumo energético en el edificio, se concluye que el algoritmo implementado por *Elevate* consigue un tiempo de tránsito bajo sacrificando el consumo energético. Esto es debido a que el algoritmo asigna muchos ascensores al mismo tiempo para lograr que los pasajeros lleguen a sus destinos sin realizar paradas intermedias. Es decir, si existen paradas intermedias que una cabina puede servir, pero la cabina no se encuentra vacía, el algoritmo prioriza que el pasajero llegue a su destino. Debido a este elevado empleo de ascensores, el consumo energético asociado a la implementación de DC-Elevate es muy elevado como se ha podido observar en el estudio realizado, siendo, por tanto, una opción poco ventajosa si se busca una gestión eficiente del edificio.

Siendo el tiempo medio de espera del pasajero frente al ascensor la minimización prioritaria debido a que la apreciación del tiempo por parte del usuario es mayor al encontrarse esperando que una vez se ha montado en la cabina. Se puede concluir con que son el algoritmo genético y la función de evaluación basada en tiempo la mejor combinación posible para la asignación de cabinas con arquitectura *Double Deck* a llamadas de pasajeros, siendo además junto con la búsqueda tabú los algoritmos que aportan un menor consumo energético.

10. REFERENCIAS

Adalid Fedriani, J. M. (2014). Heurística para la asignación de llamadas en ascensores con arquitectura *Double Deck*. Trabajo Fin de Grado.

Al-Sharif L., AlOsta E., Abualhomos N., Suhweil Y (2017). Derivation and verification of the round trip time equation and two performance coefficients for double deck elevators under incoming traffic conditions. Building services engineering research and technology, Vol 38(2), pp 176-196.

Ascensores y escaleras mecánicas Mitsubishi Electric.
http://www.enor.es/catalogo_otea_ES.pdf
Último acceso: 29/08/2018

Barney, G. (2003). Elevator traffic handbook / Spon Press.

CIBSE Guide D (2000). Transportation systems in buildings, pp 3-2.

Cortés, P., Fernández, J., Delgado, M. (2009). Controlador basado en lógica difusa para la detección del patrón de tráfico en sistemas de transporte vertical. XIII Congreso de Ingeniería de Organización. Barcelona-Terrasa, 2-4 Septiembre 2009.

Cortés, P., Larrañeta, J., Onieva, L., Muñuzuri, J., Guadix, J. (2003). Algoritmos para la Asignación de Llamadas en Sistemas de Tráfico Vertical Selectivo en Bajada. V Congreso de Ingeniería de Organización. Valladolid, 4-5 Septiembre 2003 pp. 1-4.

Glover, F. (1989). Tabu Search- Part I. ORSA Journal on Computing Vol. I, No. 3, Summer 1989.

Grupo ascensores Enor. http://www.enor.es/catalogo_otea_ES.pdf Último acceso: 29/08/2018

Ingesor. <http://www.ingesor.com/inicio/ascensores/ascensores-electromec%C3%A1nicos/>
Último acceso: 29/08/2018

La nueva era de los ascensores revoluciona la construcción de de mediana y gran altura.
<http://www.urban-hub.com/es/ideas/la-nueva-era-de-los-ascensores-revoluciona-la-construccion-de-mediana-y-gran-altura/> Último acceso: 29/08/2018

Mesa Jiménez, J. J. (2015). Sistemas de asignación de llamadas para ascensores tipo Double Deck basados en algoritmos genéticos. Trabajo Fin de Grado.

Otis en la Torre Burj Khalifa. <http://www.otis.com/site/es-esl/pages/TorreBurjKhalifa.aspx>
Último acceso: 29/08/2018

Reyes López, M. A. (2017). Optimización de los procesos de asignación de llamadas en grupos de ascensores con arquitectura double deck mediante heurísticas y metaheurísticas: análisis de tráfico interfloor. Trabajo Fin de Grado.

The Shard. <https://www.londres.es/the-shard> Último acceso: 29/08/2018

Tipos de ascensores para las edificaciones. <http://www.t3mascensores.com/tipos-de-ascensores-para-las-edificaciones/> Último acceso: 29/08/2018

Torre Picasso. <http://www.torre-picasso.com/es/torre-picasso/el-edificio>
Último acceso: 29/08/2018

Un poco de historia. http://www.silcon.com.ar/un_poco_de_historia.htm
Último acceso: 29/08/2018

Vázquez Ledesma, A. (2016). Asignación de llamadas para ascensores con arquitectura double deck mediante algoritmos de búsqueda tabú. Trabajo Fin de Grado.

ANEXO A: SOFTWARE

La implementación de los algoritmos programados requiere de un tiempo de *setup* previo en el cual es necesaria la instalación de varios programas. El primero de ellos es *Elevate 8*, software de simulación de tráfico de ascensores, desarrollado por Peters Research Ltd, en el cual se puede seleccionar el número, tipo y velocidad de ascensores en distintos tipos de edificios. En segundo lugar, se tiene el software *Microsoft Visual Studio 2008* en el cual se programan los algoritmos de resolución empleados, así como las diferentes funciones de evaluación.

Además, se debe descargar un archivo desde la propia web de *Elevate* que permite obtener la interfaz de desarrollador, que es necesaria para la conexión entre *Elevate 8* y *Visual Studio 2008*.

La instalación de *Visual Studio* en su versión 2008 es muy importante ya que además de ser el software recomendado por *Elevate*, se comprueba a través de la experiencia que es el único software que evoluciona favorablemente sin generar errores de conexión entre ambos programas.

Se pasa a detallar ahora la instalación y características de ambos softwares.

1.1. Elevate 8

Una vez se instala *Elevate 8*, se muestra una pantalla inicial como la que se observa en la Figura 175.

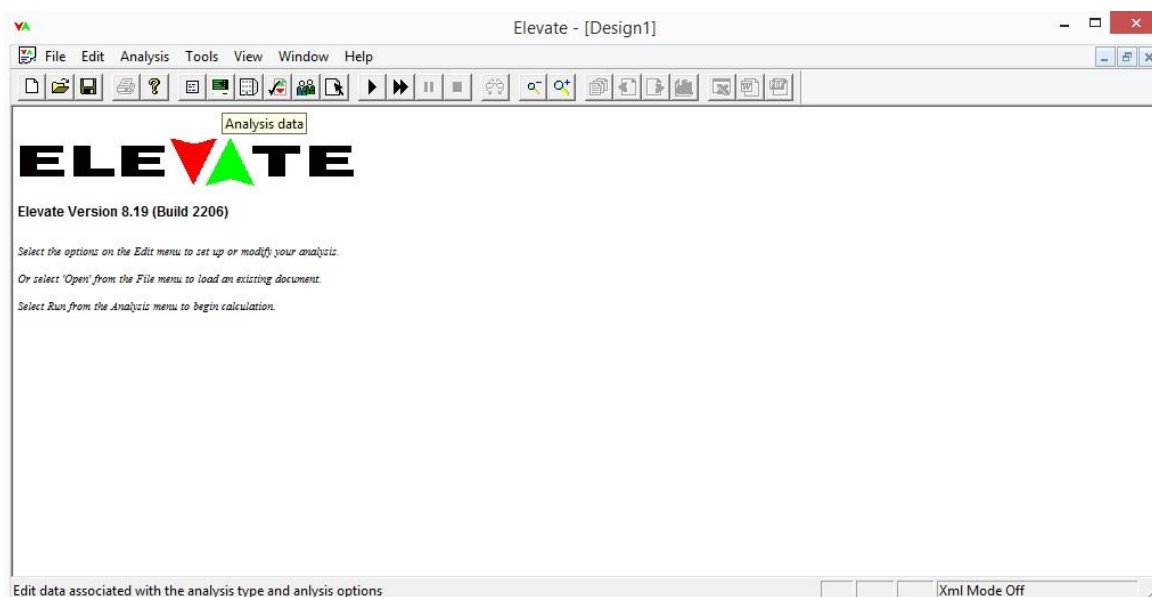


Figura 175: Ventana principal *Elevate*

1.1.1. Características del ascensor y edificio

Para configurar las características de los ascensores y edificio que se va a simular, se debe pulsar en el botón del software *Analysis Data*. Se despliega una primera ventana (se muestra en la Figura 176) en la cual se debe seleccionar en *Analysis type* la opción *Simulation* y en *Dispatcher* la opción *Custom* si se quieren obtener los algoritmos de la interfaz de desarrollador. Además, en esta ventana se puede seleccionar el número de intervalos de tiempo antes de mostrar los resultados y la duración de dichos intervalos. Además, se muestra en esta ventana la opción de realizar la simulación con el modelo energético que deriva el consumo energético del edificio.

Analysis Data

Cancel < Back Next > Finish

Analysis type: Simulation
Measurement system: ☒ Metric ☐ U.S.

Dispatcher: ☐ Standard ☒ Custom
Mode: Conventional Control example, Destination Control example, Destination Control example, Mixed Control example, Double Deck Destination Control example (selected), Double Deck Conventional Control example, Changing Speed example, Read dispatching parameters from file example

Round Trip Time calculations: Losses (%): 5

Interfaces: Mode: Xml Mode Off
Data exchange file: C:\Elevate 8\Elevate.xml
Advanced settings...

Simulation parameters: Time slice between simulation calculation (s): 0.1
No of time slices between screen updates: 10
No of simulations to run for each configuration: 10
Random number seed for passenger generator: 1

Energy: Energy model: Off
Currency: \$
Price per kWh: 0.06

Figura 176: Ventana *Analysis Data*

Building Data:

En esta ventana que se muestra en la Figura 177, se seleccionan las características del edificio tales como nombre de cada planta, altura de esta, número de pasajeros que se encuentran en cada planta y selección de la planta entrada del edificio.

The screenshot shows the 'Building Data' window. It features a table with columns: Floor Name, Floor Level (m), No of people, Area (m²), Area/person, and Entrance Floor. The table lists floors from Level 1 to Level 28. To the right of the table is a sidebar with buttons: Cancel, < Back, Next >, and Finish. Below these are buttons for 'Speed fill table', 'Clear table', 'Copy', 'Cut', and 'Paste'. Further down are radio buttons for 'Enter' (Floor height or Floor level) and 'Lowest first' or 'Highest first'. There is an 'Absenteeism (%)' input field set to 0. At the bottom, there are radio buttons for 'Express zone' or 'No express zone', and two dropdown menus for 'Lowest floor not served by elevators' and 'Highest floor not served by elevators'.

	Floor Name	Floor Level (m)	No of people	Area (m²)	Area/person	Entrance Floor
1	Level 1	0	0			<input checked="" type="checkbox"/>
2	Level 2	3,8	50			<input type="checkbox"/>
3	Level 3	7,6	50			<input type="checkbox"/>
4	Level 4	11,4	50			<input type="checkbox"/>
5	Level 5	15,2	50			<input type="checkbox"/>
6	Level 6	19	50			<input type="checkbox"/>
7	Level 7	22,8	50			<input type="checkbox"/>
8	Level 8	26,6	50			<input type="checkbox"/>
9	Level 9	30,4	50			<input type="checkbox"/>
10	Level 10	34,2	50			<input type="checkbox"/>
11	Level 11	38	50			<input type="checkbox"/>
12	Level 12	41,8	50			<input type="checkbox"/>
13	Level 13	45,6	50			<input type="checkbox"/>
14	Level 14	49,4	50			<input type="checkbox"/>
15	Level 15	53,2	50			<input type="checkbox"/>
16	Level 16	57	50			<input type="checkbox"/>
17	Level 17	60,8	50			<input type="checkbox"/>
18	Level 18	64,6	50			<input type="checkbox"/>
19	Level 19	68,4	50			<input type="checkbox"/>
20	Level 20	72,2	50			<input type="checkbox"/>
21	Level 21	76	50			<input type="checkbox"/>
22	Level 22	79,8	50			<input type="checkbox"/>
23	Level 23	83,6	50			<input type="checkbox"/>
24	Level 24	87,4	50			<input type="checkbox"/>
25	Level 25	91,2	50			<input type="checkbox"/>
26	Level 26	95	50			<input type="checkbox"/>
27	Level 27	98,8	50			<input type="checkbox"/>
28	Level 28	102,6	50			<input type="checkbox"/>
29						<input type="checkbox"/>
30						<input type="checkbox"/>
31						<input type="checkbox"/>
32						<input type="checkbox"/>
33						<input type="checkbox"/>
34						<input type="checkbox"/>

Figura 177: Ventana Building Data

Elevator Data:

En esta ventana, se seleccionan las características de los ascensores. Se pueden configurar en modo *Standard* en dónde se modifican las características de todos los ascensores a la vez o modo *Advanced* en cuyo caso se modifican las características de cada ascensor por independiente. Se muestran en la Figura 178 y Figura 179 ambos modos de configuración.

The screenshot shows the 'Elevator Data' window in 'Standard' mode. It has a sidebar with 'Cancel', '< Back', 'Next >', and 'Finish' buttons. The main area contains various configuration options: 'Elevator selection mode' (Standard selected), 'No. of Elevators' (Specified, 7), 'Type' (Single Deck), 'Capacity (kg)' (Specified, 1000), 'Car Area (m²)' (Auto), 'Door times (s)' (Auto), 'Door dwell (s)' (Dwell 1: 3, Dwell 2: 2), 'Speed (m/s)' (Specified, 2.5), 'Acceleration (m/s²)' (Auto), 'Jerk (m/s³)' (Auto), 'Start delay (s)' (0.5), 'Levelling delay (s)' (0), 'Home floor' (Level 1), and 'Destination Call Stations' (select). There are also 'Min' and 'Max' value fields for several parameters.

Figura 178: Ventana standard Elevator Data

Elevator Data

Elevator selection mode
☐ Standard ☒ Advanced

Cancel < Back Next > Finish

	Car 1	Car 2	Car 3	Car 4	Car 5	Car 6	Car 7
Type	Single Deck	Single Deck	Single Deck	Single Deck	Single Deck	Single Deck	Single Deck
Capacity (kg)	1000	1000	1000	1000	1000	1000	1000
Floor Area (m²)	2.4	2.4	2.4	2.4	2.4	2.4	2.4
Door pre-opening (s)	0	0	0	0	0	0	0
Door open time (s)	1.8	1.8	1.8	1.8	1.8	1.8	1.8
Door close time (s)	2.9	2.9	2.9	2.9	2.9	2.9	2.9
Home Door dwell 1 (s)	3	3	3	3	3	3	3
Home Door dwell 2 (s)	2	2	2	2	2	2	2
Door dwell 1 (s)	3	3	3	3	3	3	3
Door dwell 2 (s)	2	2	2	2	2	2	2
Max door re-openings	unlimited	unlimited	unlimited	unlimited	unlimited	unlimited	unlimited
Speed (m/s)	2.5	2.5	2.5	2.5	2.5	2.5	2.5
Acceleration (m/s²)	0.8	0.8	0.8	0.8	0.8	0.8	0.8
Jerk (m/s³)	1.6	1.6	1.6	1.6	1.6	1.6	1.6
Start Delay (s)	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Levelling Delay (s)	0	0	0	0	0	0	0
Home Floor	Level 1	Level 1	Level 1	Level 1	Level 1	Level 1	Level 1
shut down time (s)	0	0	0	0	0	0	0
restart time (s)	0	0	0	0	0	0	0
Service	Auto	Auto	Auto	Auto	Auto	Auto	Auto

Configuration Floors Served

Copy Cut Paste

Figura 179: Ventana *advanced Elevator Data*

Passenger Data:

En esta ventana se seleccionan las características del tráfico de pasajeros del edificio. Al igual que en la ventana anterior, se cuenta con un modo *Standard* para la selección de características para toda la población por igual y un modo *Advanced* para una configuración personalizada a cada periodo de tiempo. Se pueden observar ambas ventanas en la Figura 180 y Figura 181.

Passenger Data

Passenger Data Mode
☒ Standard ☐ Advanced ☐ File

Cancel < Back Next > Finish

Traffic generator

Arrangement: Double Deck with no odd to even floor traffic

Template: Constant traffic (% building pop per 5 mins)

First floor served upper zone: Level 1

Demand (% pop per 5 mins): 15

% Incoming: 30 % Outgoing: 30 % Interfloor: 40

Min demand (% pop per 5 mins): 1 Max demand (% pop per 5 mins): 15

Step duration (mins): 5 Step height (%): 1

Start Time: 11 hrs 0 mins End Time: 13 hrs 15 mins

Passenger details

Loading Time (s): 1.2 Unloading Time (s): 1.2

Passenger Mass (kg): 75 Capacity Factor by Mass (%): 80

Passenger Area (m²): 0.21 Capacity Factor by Area (%): 100

Stair Factor (%): 0

Entrance level bias

	% bias
Level 1 & Level 2	100

Copy Cut Paste

Figura 180: Ventana *standard Passenger Data*

Passenger Data

Passenger Data Mode
☐ Standard ☒ Advanced ☐ File

	Start Time (hrs:mins)	End Time (hrs:mins)	Passenger Mass (kg)	Capacity Factor by Mass (%)	Passenger Area (m ²)	Capacity Factor by Area (%)	Loading Time (s)	Unloading Time (s)	Stair Factor (%)	Notes
Period 1	11:00	13:15	75	80	0,21	100	1,2	1,2	0	Passengers
Period 2										
Period 3										
Period 4										
Period 5										
Period 6										
Period 7										
Period 8										
Period 9										
Period 10										
Period 11										
Period 12										
Period 13										
Period 14										
Period 15										
Period 16										
Period 17										
Period 18										
Period 19										
Period 20										
Period 21										
Period 22										
Period 23										
Period 24										
Period 25										
Period 26										
Period 27										

Figura 181: Ventana *advanced Passenger Data*

Se pueden seleccionar varios tipos de tráfico de pasajeros, siendo los principales:

- **CIBSE:**

Se caracteriza por tener un periodo *uppeak* abrupto al inicio de la jornada y un periodo *downpeak* fuerte al final de la jornada. Al medio día tiene dos periodos de carácter moderado, uno *downpeak* y otro *uppeak*. Se muestra dicho patrón en la Figura 182 (CIBSE Guide, 2000).

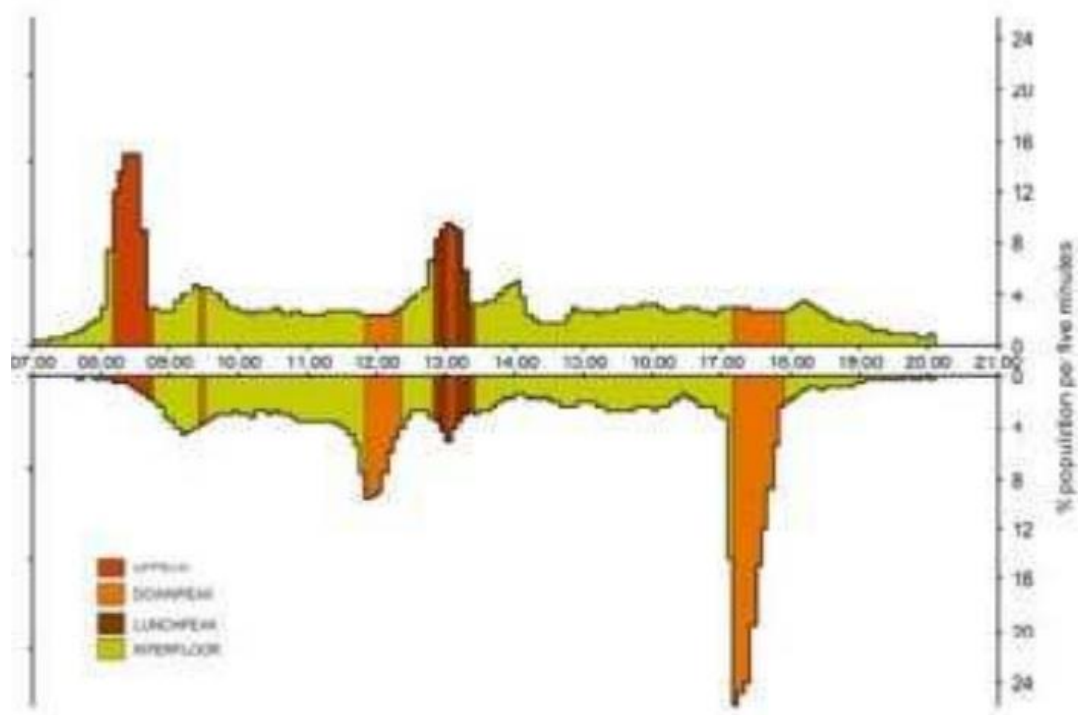


Figura 182: Patrón de tráfico CIBSE

- **Strackosh:**

Se caracteriza por ser un modelo mucho más atípico, con curvas suaves, máximos pequeños y periodos ligeros *uppeak* y *downpeak* durante toda la mañana y tarde. Se muestra en la Figura 183.

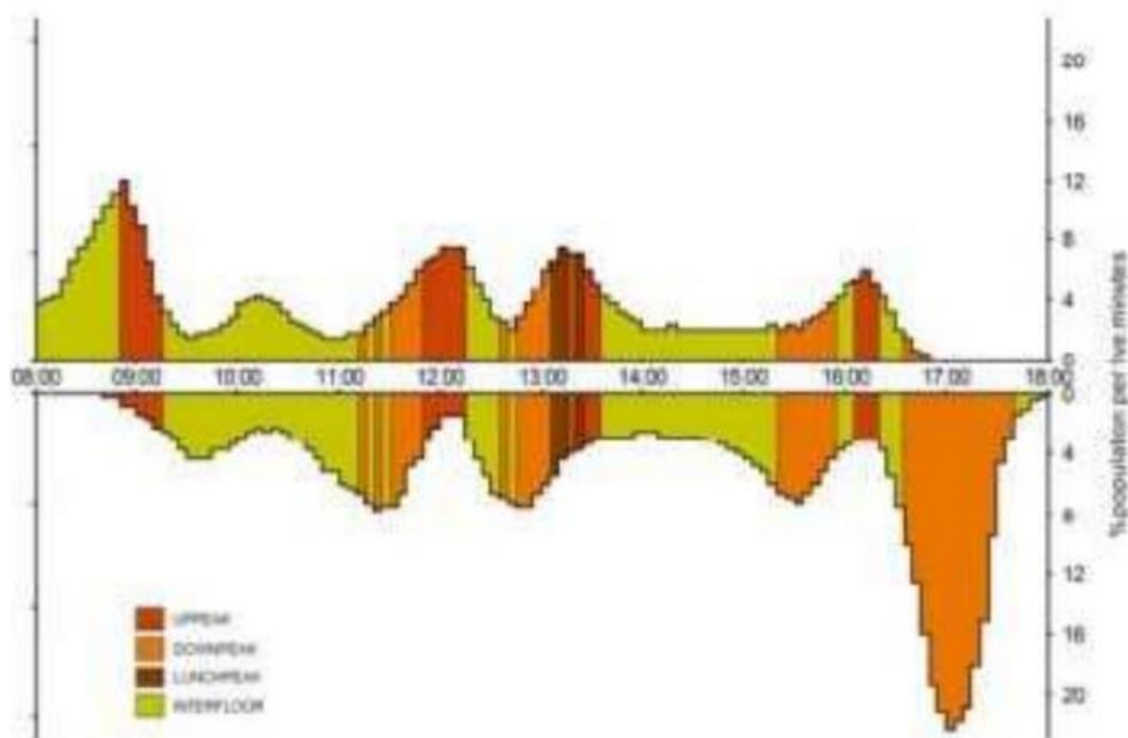


Figura 183: Patrón de tráfico *Strackosh*

- **Siikonen:**

Se caracteriza por ser el patrón de tráfico que más se aproxima al comportamiento real de flujo de pasajeros. Se muestra en la Figura 184.

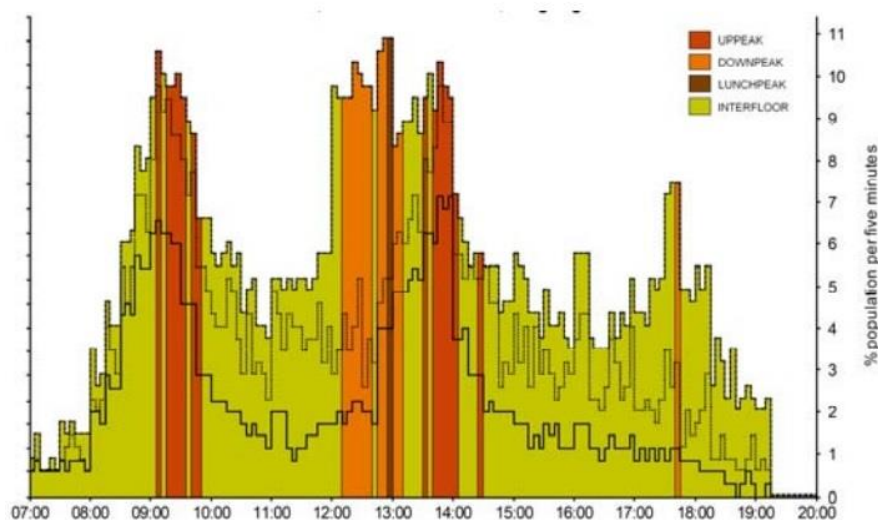


Figura 184: Patrón de tráfico *Siikonen*

Report options:

En esta ventana que se puede observar en la Figura 185, se seleccionan los tipos de análisis que se quiere que se desplieguen automáticamente al finalizar la simulación.

Report options

Cancel < Back Next > Finish

5 min Handling Capacity: filter off 12 (%) Interval: filter off 30 (s)

Capacity Factor by Mass: filter off 80 (%) Capacity Factor by Area: filter off 100 (%)

Average Waiting Time: filter off 20 (s) Average Transit Time: filter off 90 (s)

Average Time to Destination: filter off 110 (s) Maximum queue: filter off 50 (% population)

Show results for: Average of all runs All passengers 00:00 and 23:59

Graph type: ☐ Black and White ☒ Colour

	Summary	Level 1 or car 1	Level 2 or car 2	Level 3 or car 3	Level 4 or car 4	Level 5 or car 5	Level 6 or car 6	Level 7 or car 7	Level 8
Passenger Demand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Total Passenger Activity	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Passenger Transfer by Floor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Queue Lengths	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Spatial Plot	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Car Loading on Departure from Home Floor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Car Loading on Arrival at Home Floor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Dispatch Interval from Home Floor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Average Waiting and Time to Destination	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Distribution of Passenger Waiting Times	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Distribution of Passenger Transit Times	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Distribution of Time to Destination	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Car Service	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Energy Consumption	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Graphs to plot Scales

Figura 185: Ventana *Report Options*

Simulación

Una vez se ha configurado lo anterior, se puede visualizar una simulación del edificio y ascensores. En esta ventana se encuentran las diferentes plantas del edificio, los ascensores con sus respectivas puertas, flechas de color verde y rojo para las llamadas que se realizan de cada planta (ascendentes y descendentes, respectivamente) y puntos azules que indican los destinos de los pasajeros. Se muestra dicha ventana en la Figura 186.

Elevate - [Design1]

File Edit Analysis Tools View Window Help

Time (hrs:min:sec) 11:03:24 Direction ^ ^

AWT (s) 32.9 Position (m) 31.37 22.80

ATT (s) 40.7 Speed (m/s) 1.24 0.00

Load (kg) 75:150 300:150

Floor Name	People Waiting	Landing Calls	Car 1	Car 2
Level 12	2			
Level 11	1			
Level 10	2			
Level 9	4			
Level 8	0			
Level 7	2			
Level 6	2			
Level 5	0			
Level 4	2			
Level 3	0			
Level 2	3			
Level 1	1			

File Name/Location: Currently running configuration
Run number for this configuration: 1 of 10

2 No. 1000 kg elevators @ 2.50 m/s

NUM | Xml Mode Off

Figura 186: Ventana de simulación de *Elevate*

Resultados del software

Finalizada la simulación, *Elevate* ofrece los resultados que se han seleccionado en la ventana *Report Options*. Para este trabajo son de especial utilidad los resultados obtenidos del tiempo medio de espera, tiempo medio de tránsito y consumo energético acumulado.

1.2. Microsoft Visual Studio 2008

Se requiere la instalación de este programa ya que es el recomendado por *Elevate* para el desarrollo del algoritmo a través de la interfaz de desarrollador. La instalación de dicha interfaz no es trivial debido a que se requiere exactamente la versión 2008 del software para el correcto funcionamiento de la conexión entre ambos softwares a través de la interfaz.

1.2.1. Interfaz de desarrollador

Al descargar el archivo necesario para la instalación de la interfaz de desarrollador de la web de Peters Research Ltd. hay que seguir una serie de pasos detallados en una carpeta llamada *How to Use*. Hay un aspecto a destacar, y es que a la hora de la configuración del vinculador siguiendo los pasos *Propiedades* → *Todas las configuraciones* → *Vinculador* se debe generar el archivo “Dispatch_00.dll” en una carpeta que no exija permisos de administrador para evitar problemas de depuración. Se muestra en la Figura 187 la ventana de propiedades de *Visual Studio* en la cual habrá que realizar los cambios pertinentes para conseguir la interfaz de desarrollador.

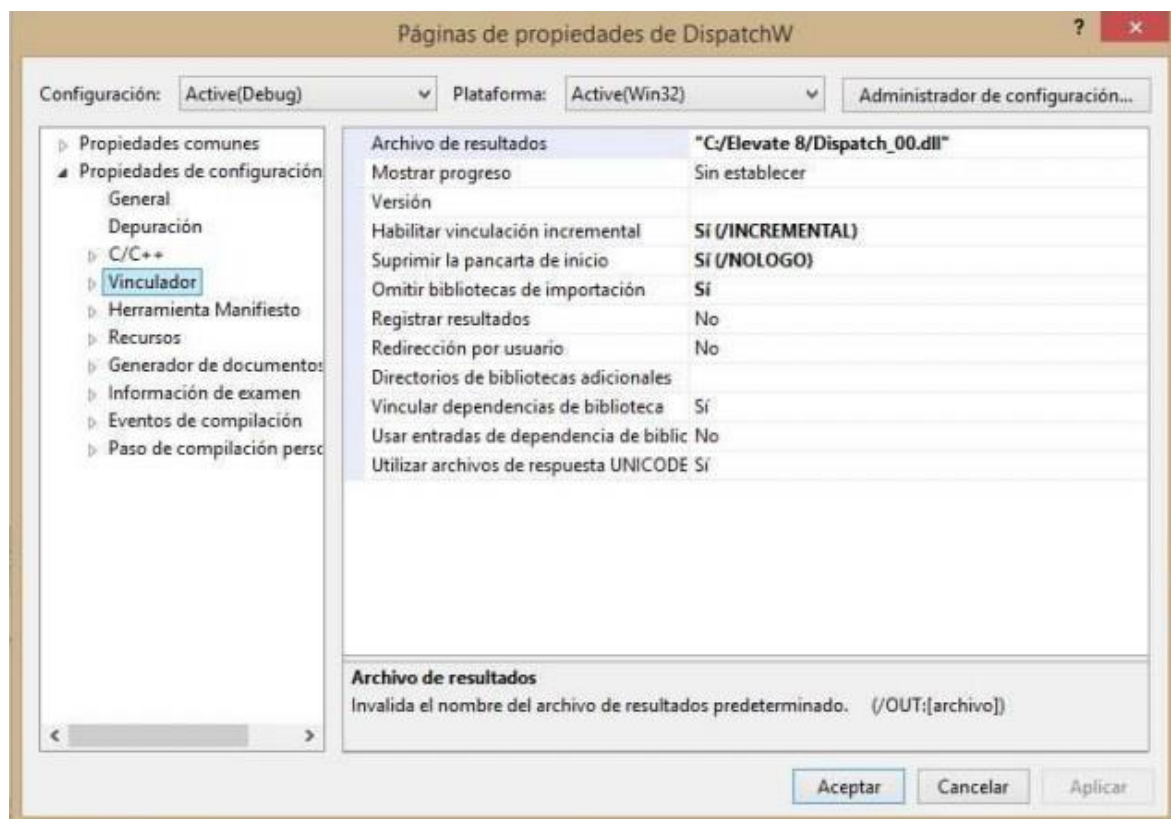


Figura 187: Ventana de propiedades de *Microsoft Visual Studio*

1.2.2. Desarrollo en Microsoft Visual Studio

El lenguaje de programación que se emplea para la programación del algoritmo es C++. Se muestra en la Figura 188 parte del algoritmo desarrollado en *Microsoft Visual Studio*.

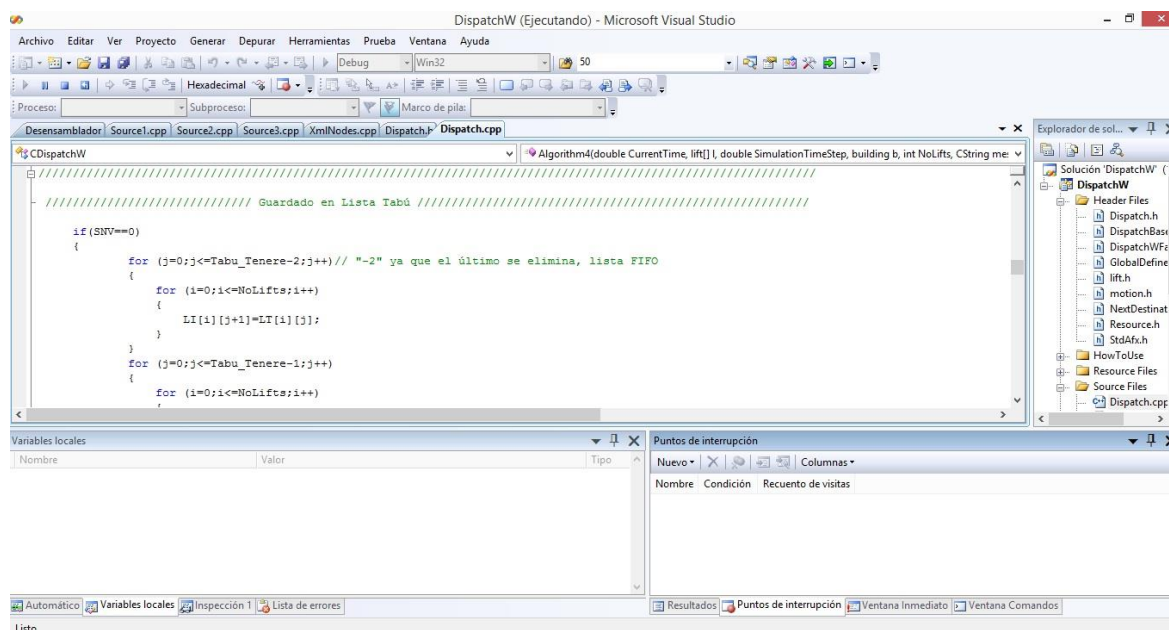


Figura 188: Extracto del algoritmo programado en Visual Studio

A través del empleo de puntos de rotura en el código se consigue depurar para observar que el algoritmo programado realiza la función requerida además de poder extraer los valores que toman las variables del problema en cada instante.

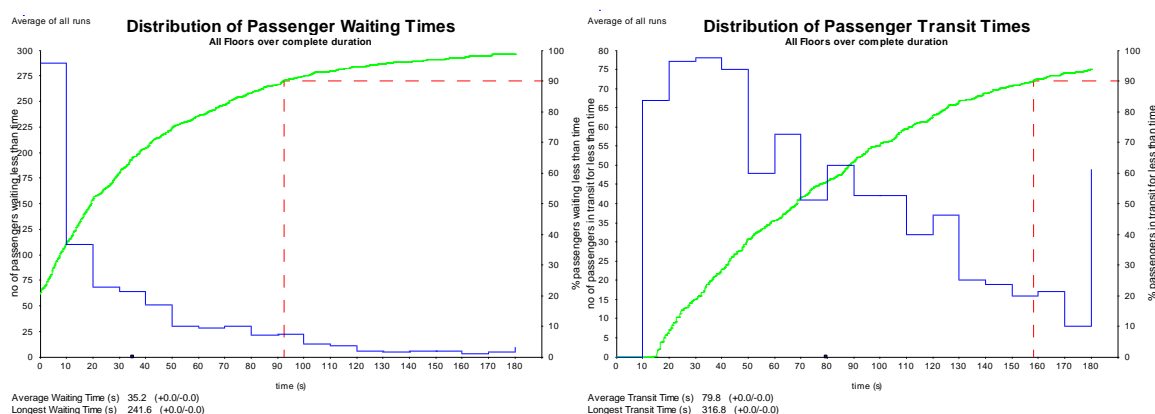
ANEXO B: GRÁFICAS EXPERIMENTALES

Este Anexo B contiene las gráficas experimentales obtenidas para el tiempo medio de espera de los pasajeros frente al ascensor y el tiempo medio de tránsito que *Elevate 8* ofrece para cada una de las simulaciones realizadas. Se muestran las gráficas agrupadas según el algoritmo empleado (búsqueda tabú y algoritmo genético) para la asignación de llamadas a ascensores, tipo de tráfico y función de evaluación.

Interfloor- Función de evaluación basada en distancia

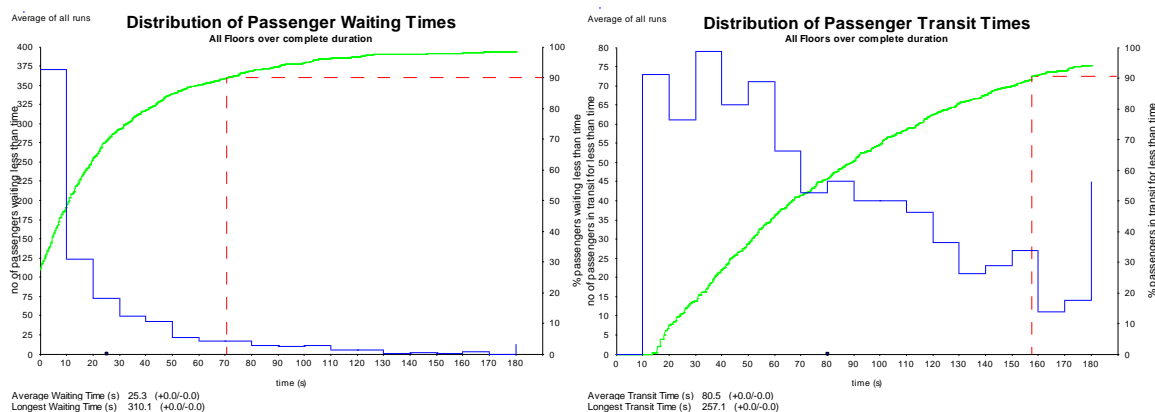
Búsqueda Tabú

36 plantas, 16 ascensores



Algoritmo Genético

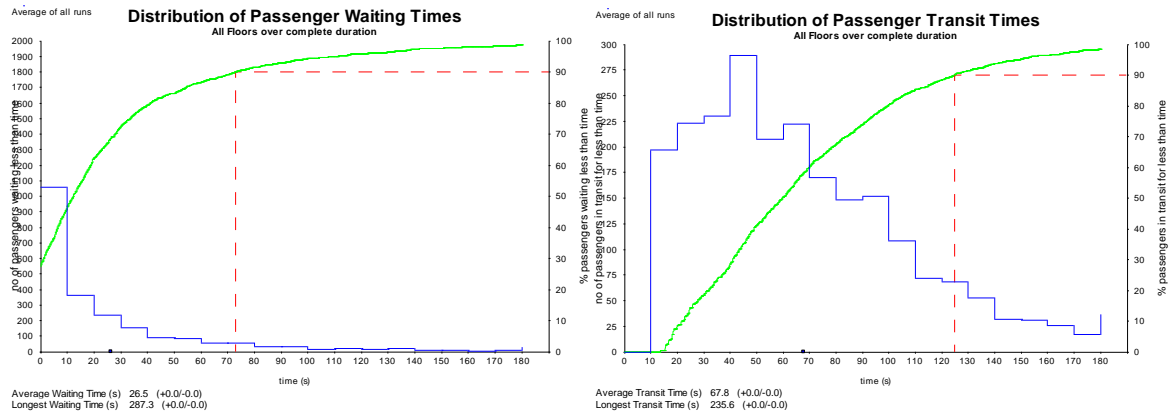
36 plantas, 16 ascensores



Lunchpeak- Función de evaluación basada en distancia

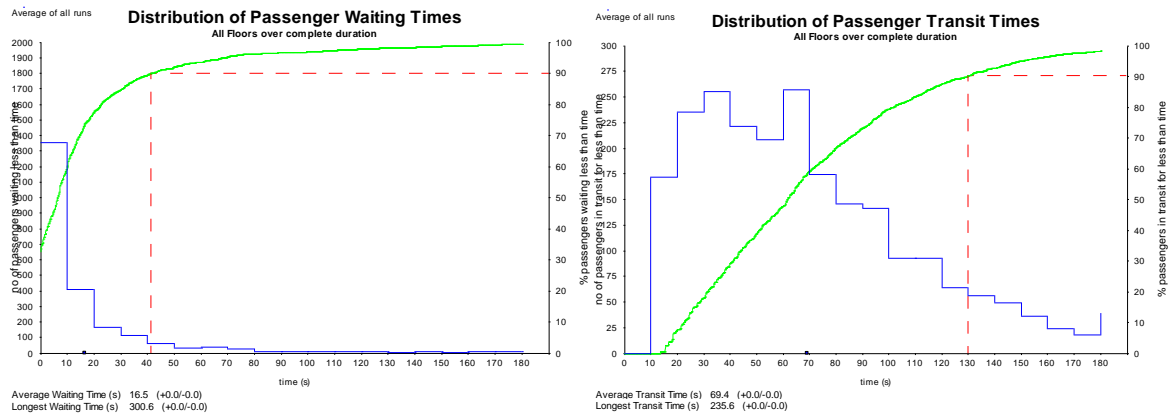
Búsqueda Tabú

36 plantas, 16 ascensores



Algoritmo Genético

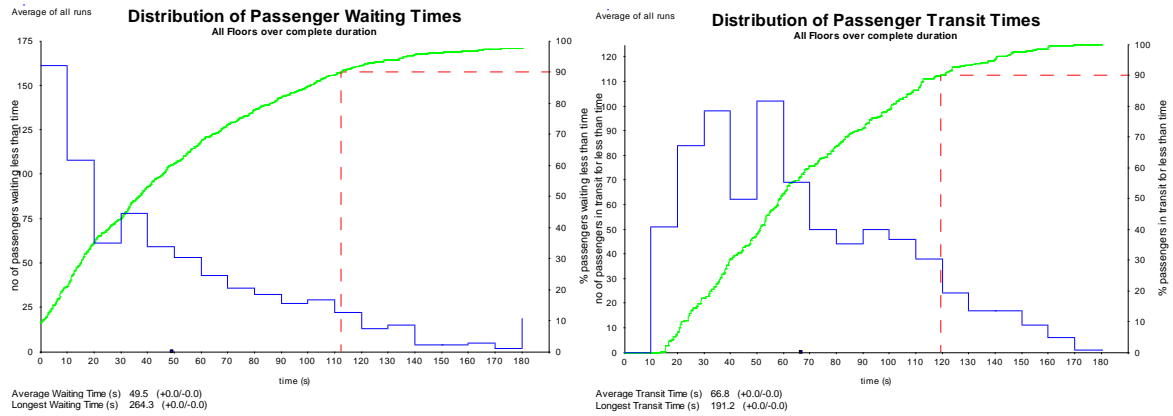
36 plantas, 16 ascensores



Downpeak- Función de evaluación basada en distancia

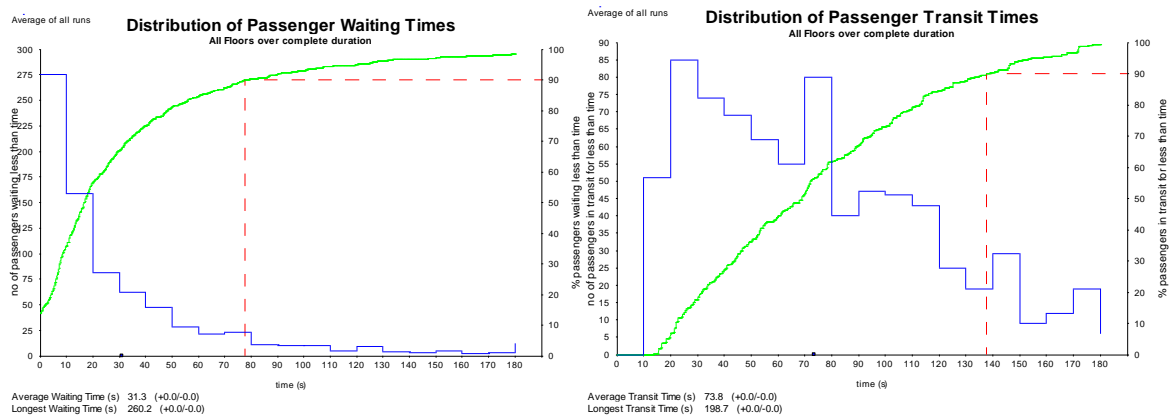
Búsqueda Tabú

36 plantas, 16 ascensores



Algoritmo Genético

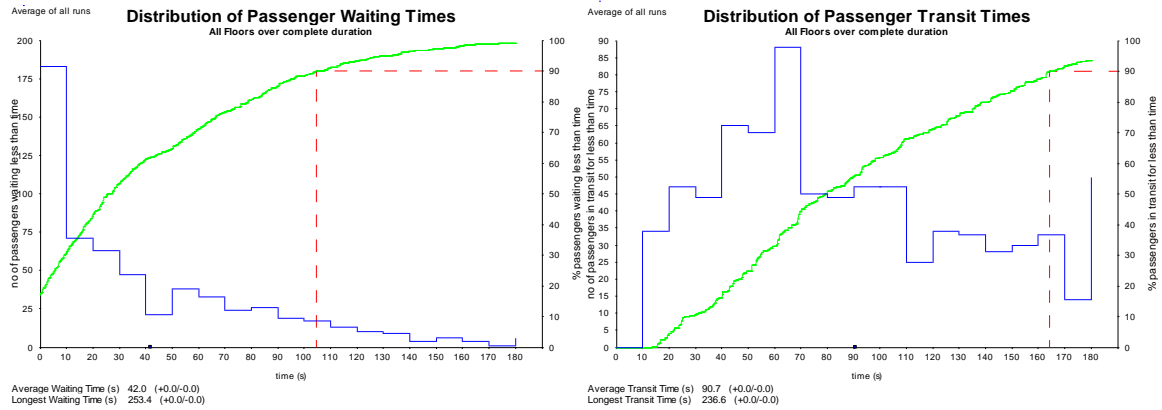
36 plantas, 16 ascensores



Interfloor- Función de evaluación basada en tiempo

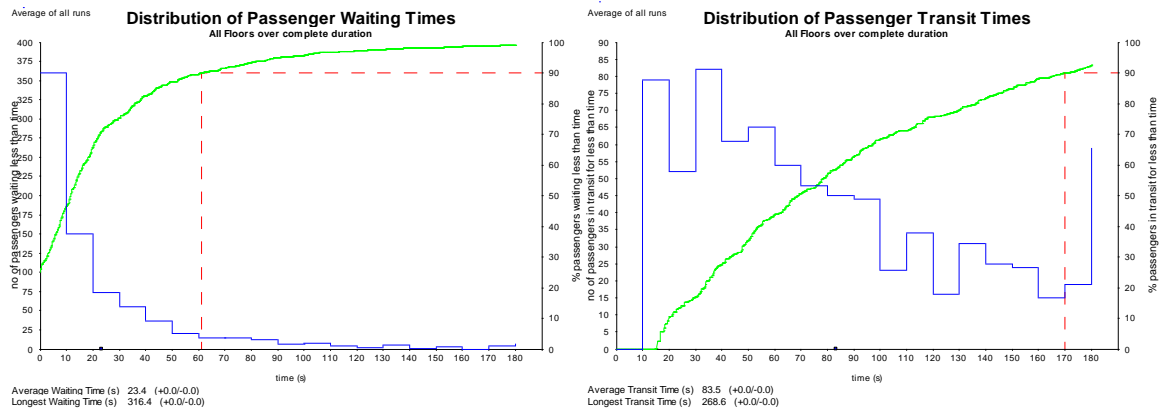
Búsqueda Tabú

36 plantas, 16 ascensores



Algoritmo Genético

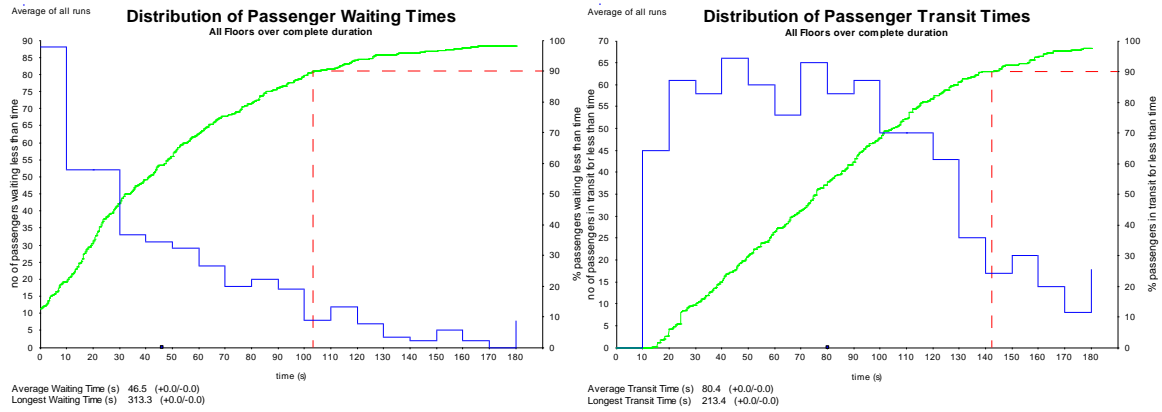
36 plantas, 16 ascensores



Lunchpeak- Función de evaluación basada en tiempo

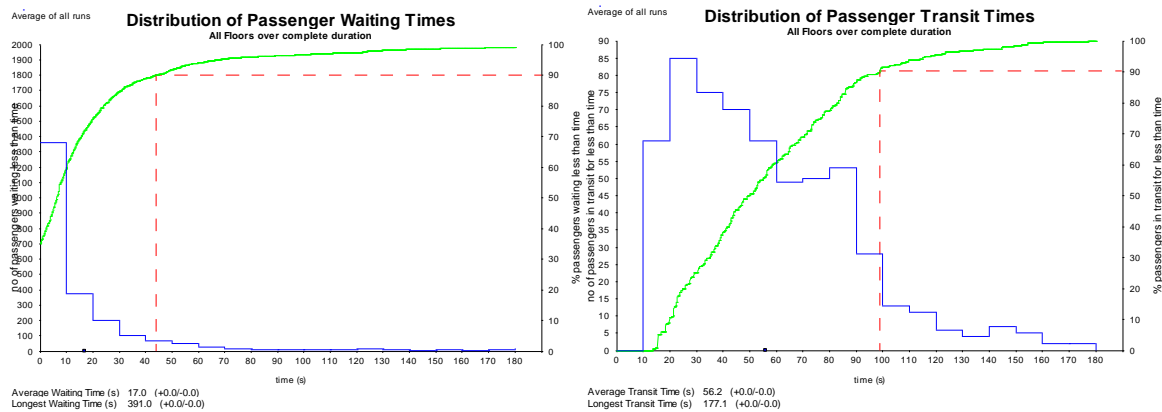
Búsqueda Tabú

36 plantas, 16 ascensores



Algoritmo Genético

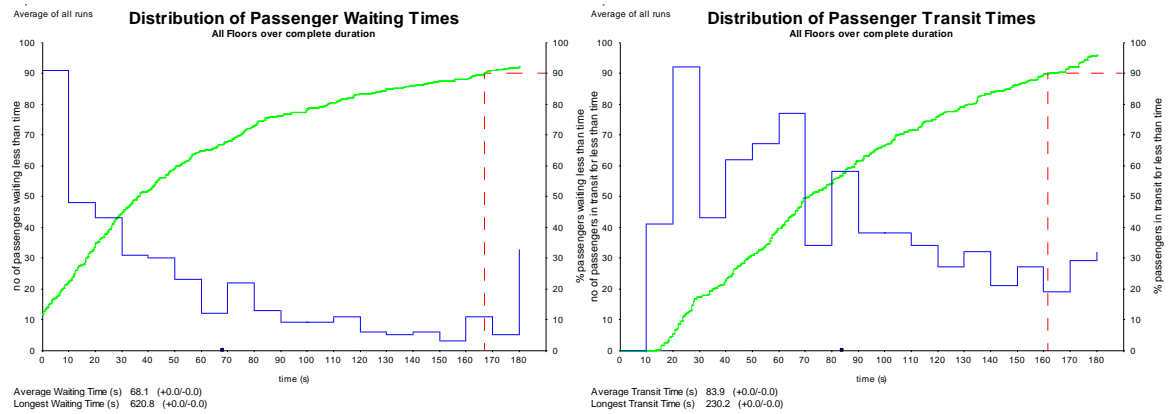
36 plantas, 16 ascensores



Downpeak- Función de evaluación basada en tiempo

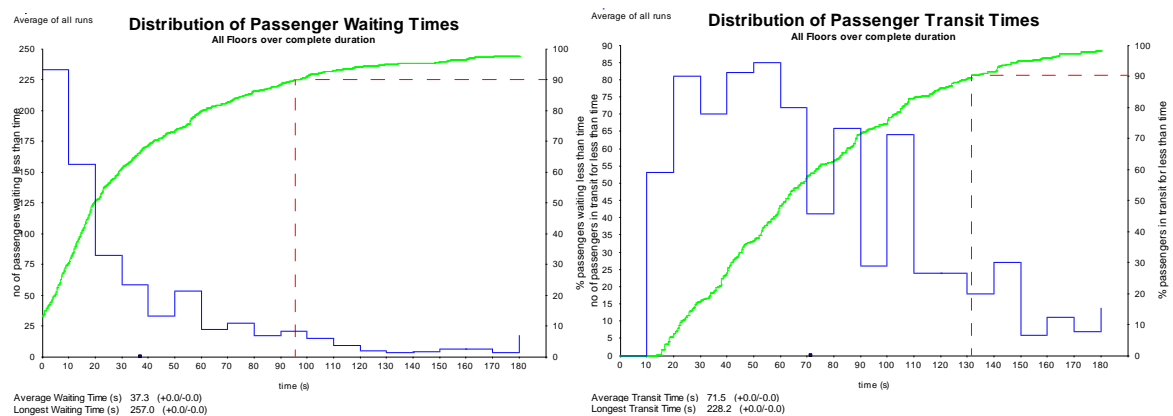
Búsqueda Tabú

36 plantas, 16 ascensores



Algoritmo Genético

36 plantas, 16 ascensores

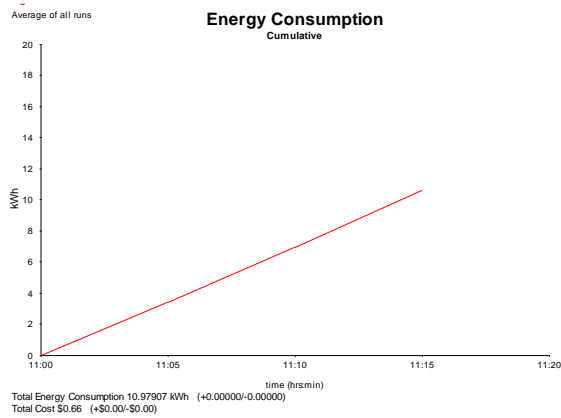


Se muestran a continuación las gráficas experimentales obtenidas del estudio del consumo energético acumulado del ascensor para cada uno de los tráficos, métodos de resolución, función de evaluación y combinación del número de plantas/ascensores.

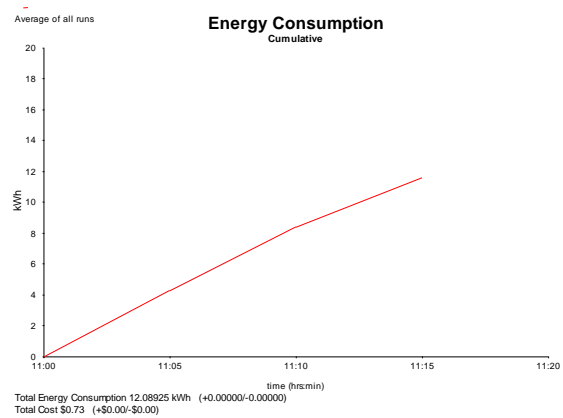
Interfloor- Función de evaluación basada en distancia

Búsqueda Tabú

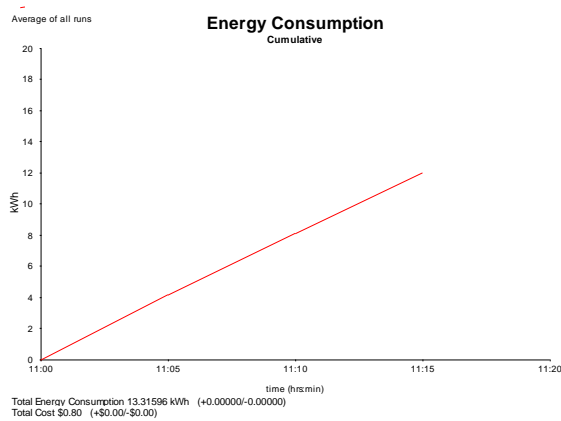
12 plantas, 5 ascensores



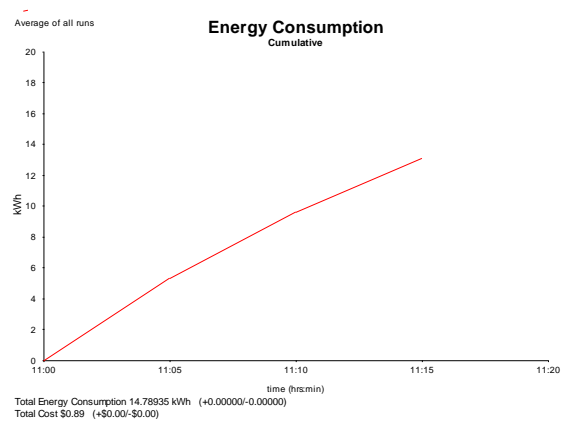
20 plantas, 8 ascensores



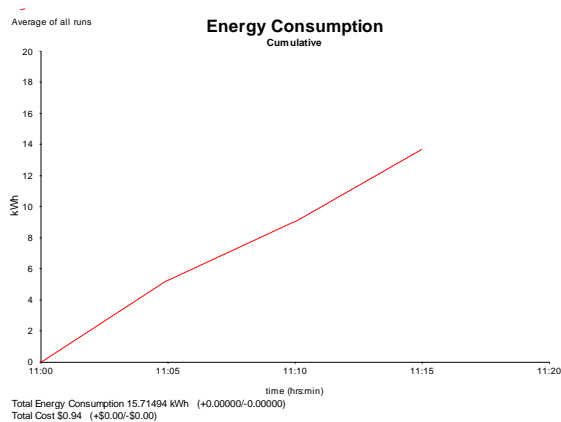
24 plantas, 10 ascensores



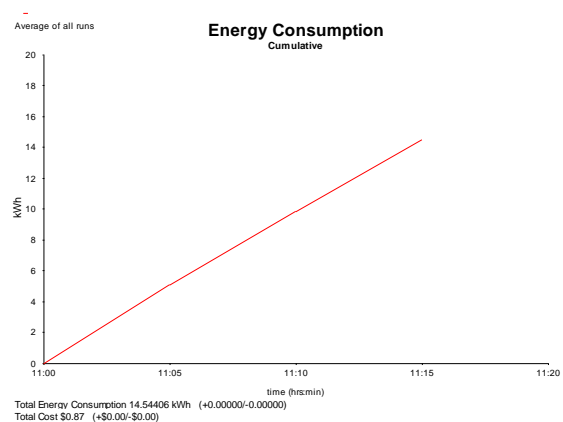
28 plantas, 13 ascensores



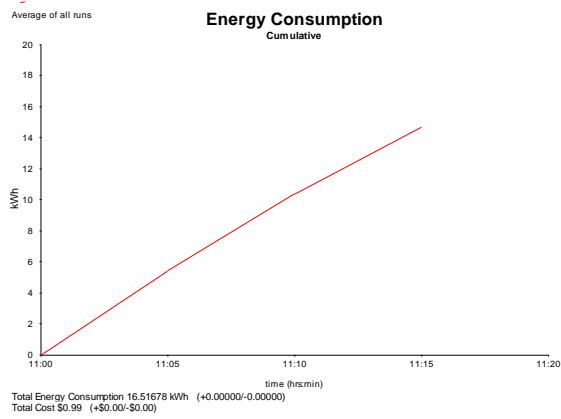
32 plantas, 14 ascensores



36 plantas, 16 ascensores

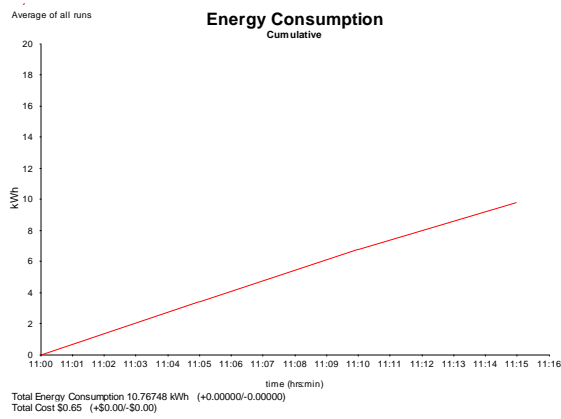


40 plantas, 19 ascensores

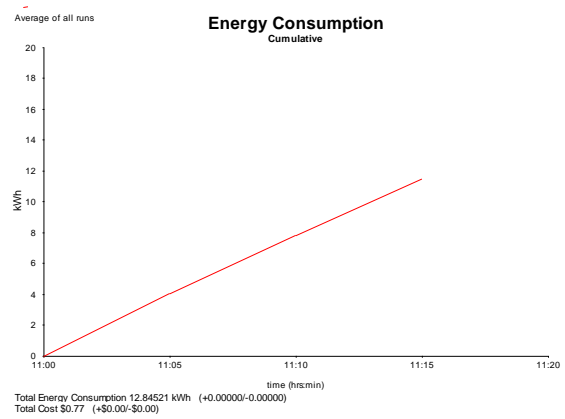


Algoritmo Genético

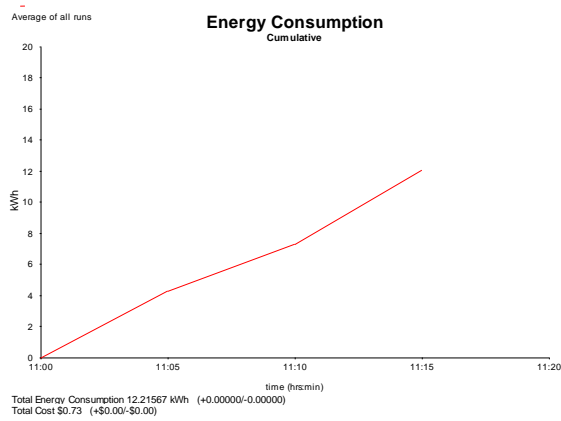
12 plantas, 5 ascensores



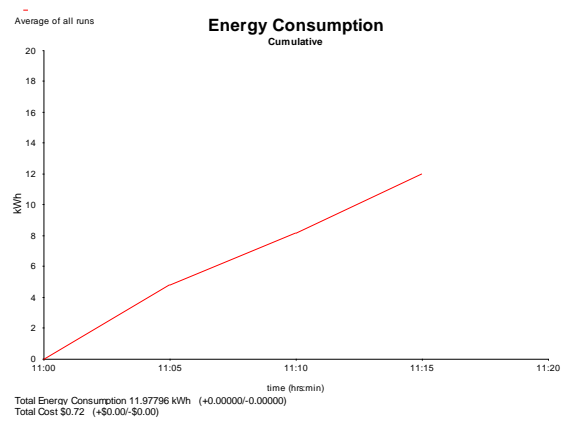
20 plantas, 8 ascensores



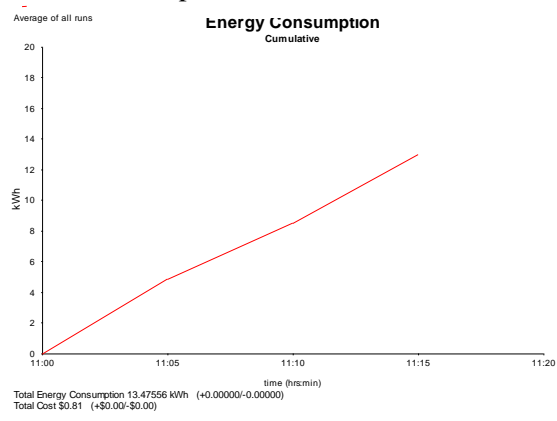
24 plantas, 10 ascensores



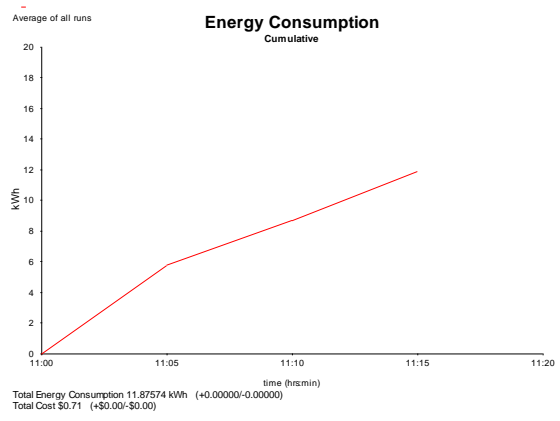
28 plantas, 13 ascensores



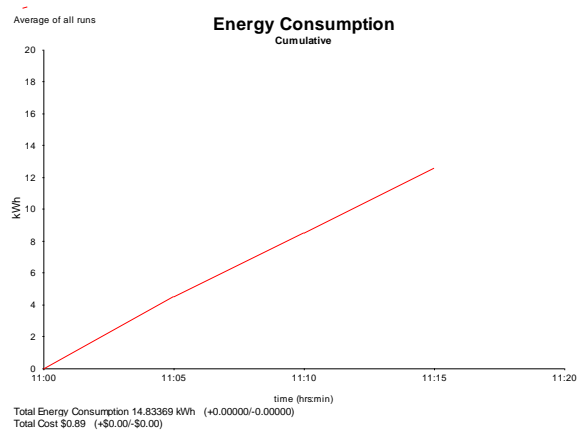
32 plantas, 14 ascensores



36 plantas, 16 ascensores



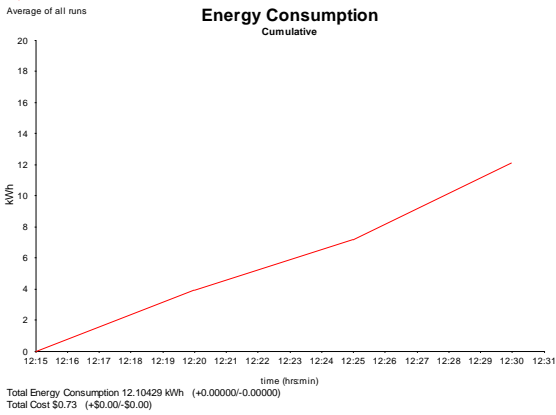
40 plantas, 19 ascensores



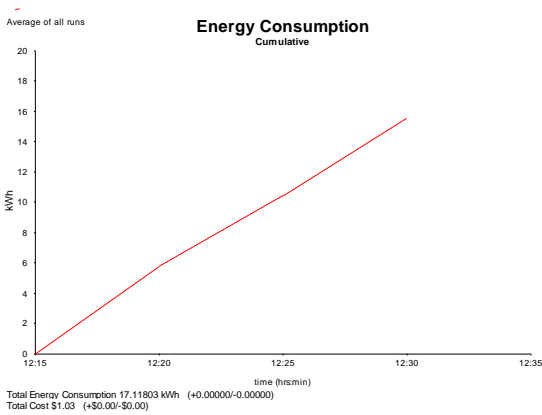
Lunchpeak- Función de evaluación basada en distancia

Búsqueda Tabú

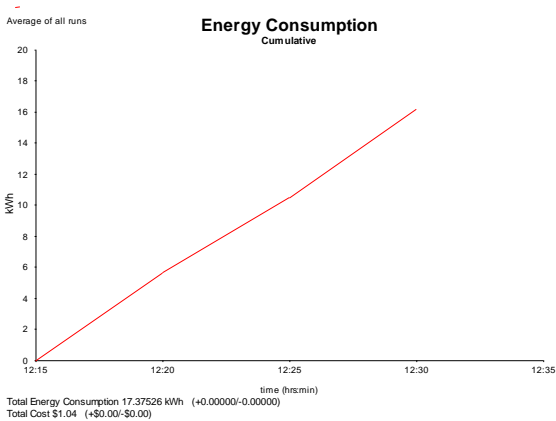
12 plantas, 5 ascensores



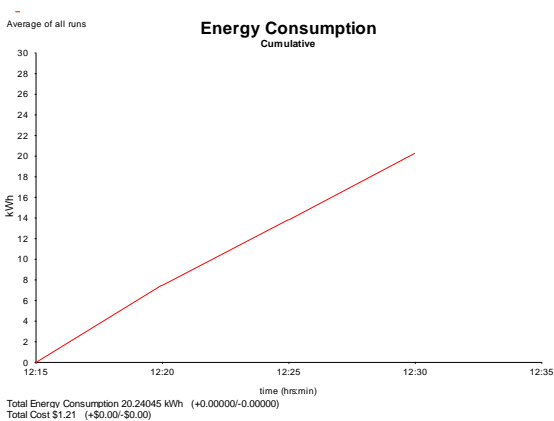
20 plantas, 8 ascensores



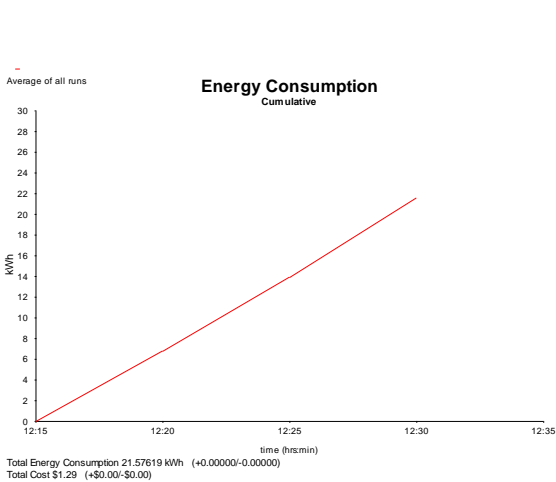
24 plantas, 10 ascensores



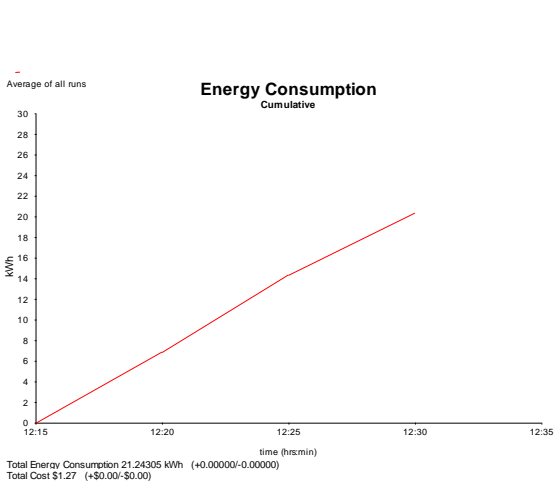
28 plantas, 13 ascensores



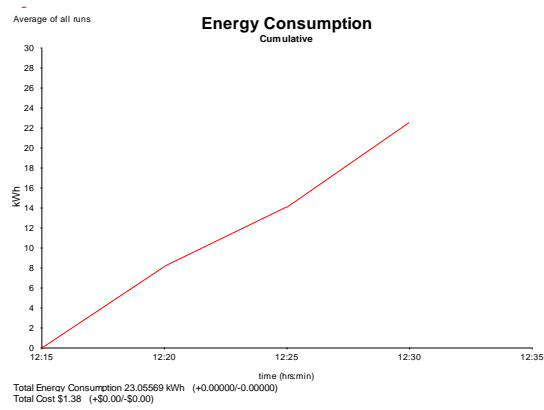
32 plantas, 14 ascensores



36 plantas, 16 ascensores

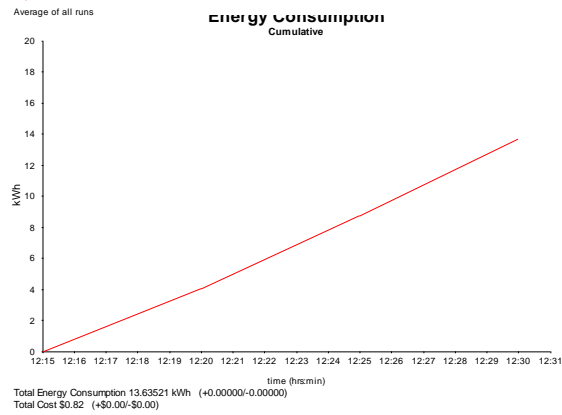


40 plantas, 19 ascensores

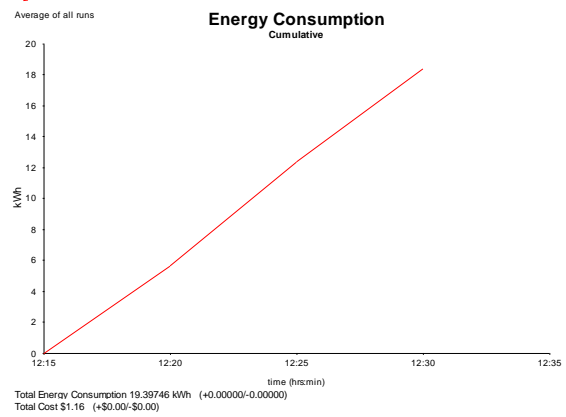


Algoritmo Genético

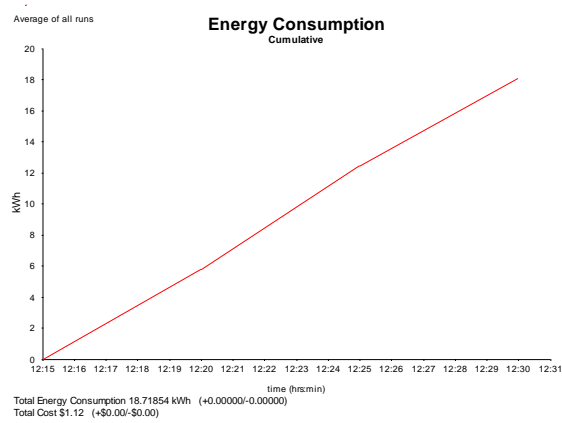
12 plantas, 5 ascensores



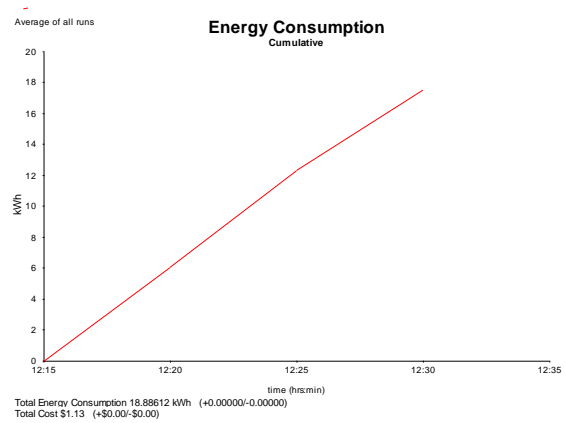
20 plantas, 8 ascensores



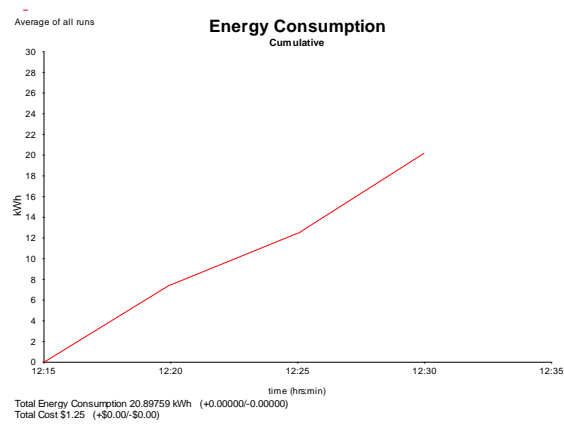
24 plantas, 10 ascensores



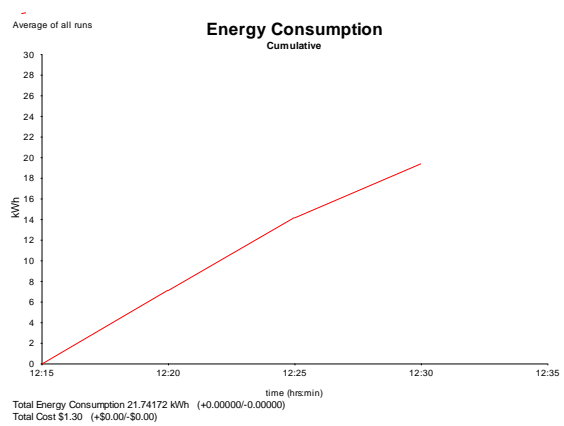
28 plantas, 13 ascensores



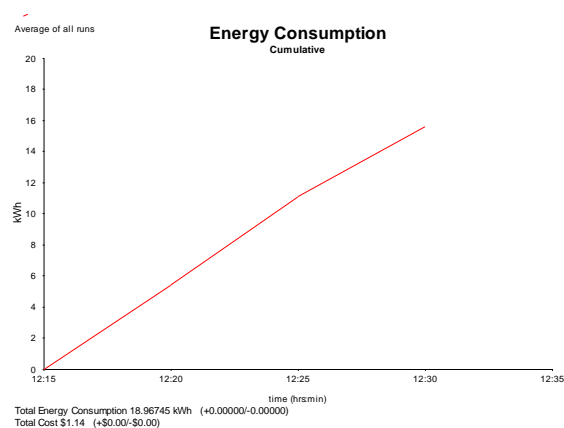
32 plantas, 14 ascensores



36 plantas, 16 ascensores



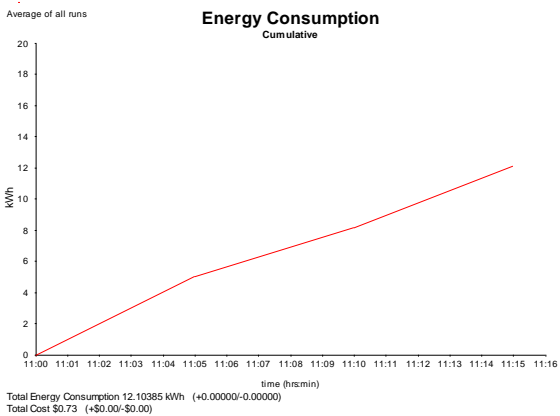
40 plantas, 19 ascensores



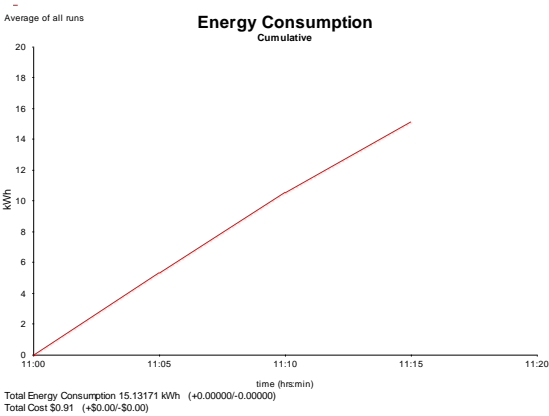
Downpeak- Función de evaluación basada en distancia

Búsqueda Tabú

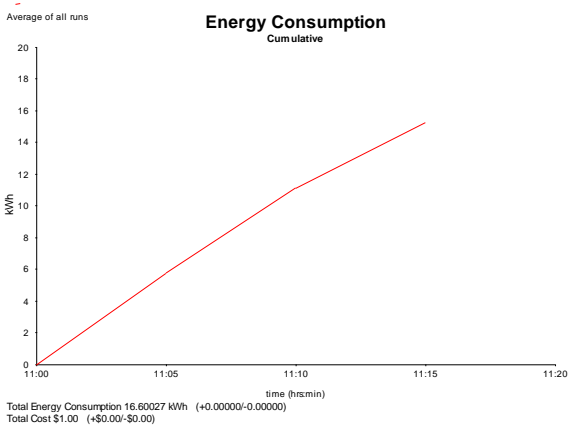
12 plantas, 5 ascensores



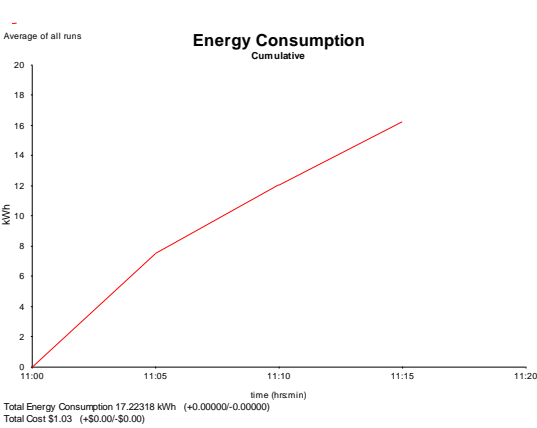
20 plantas, 8 ascensores



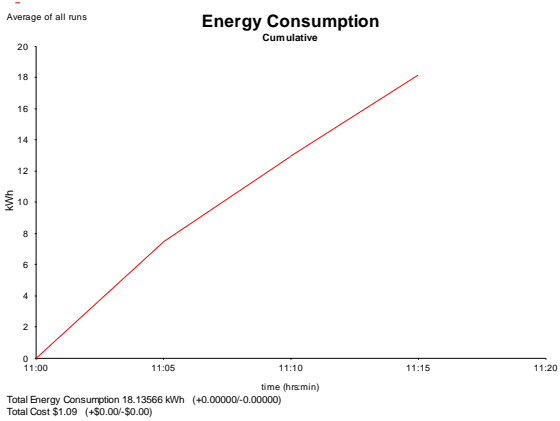
24 plantas, 10 ascensores



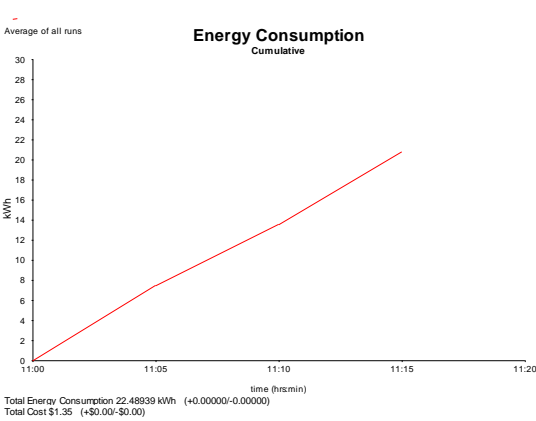
28 plantas, 13 ascensores



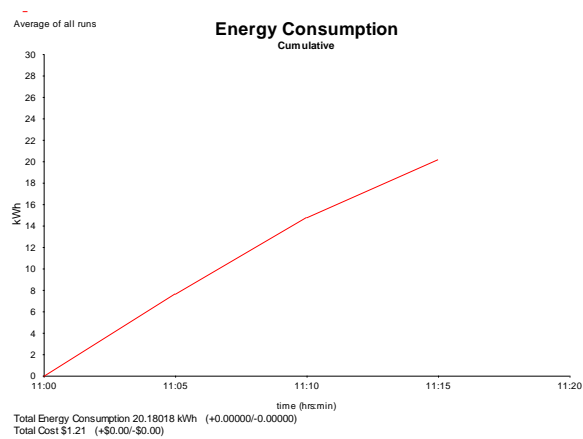
32 plantas, 14 ascensores



36 plantas, 16 ascensores

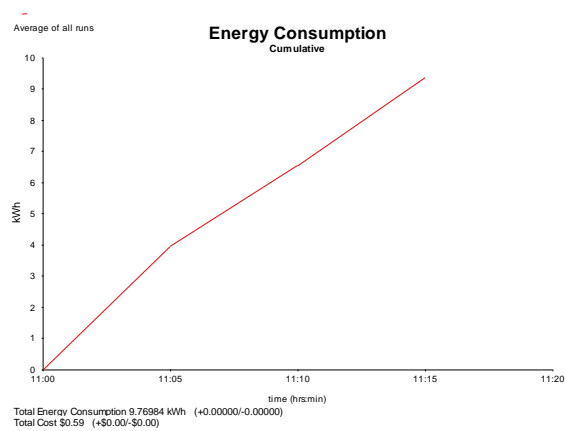


40 plantas, 19 ascensores

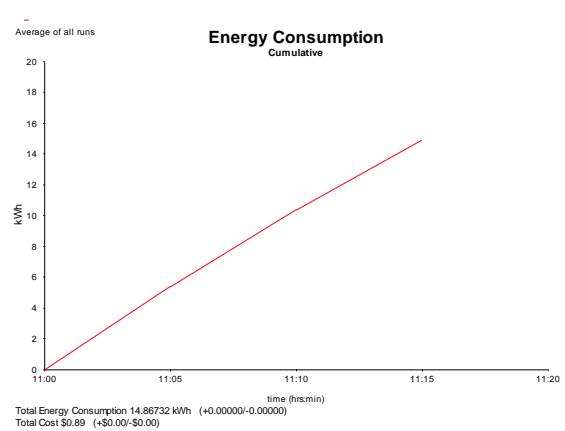


Algoritmo Genético

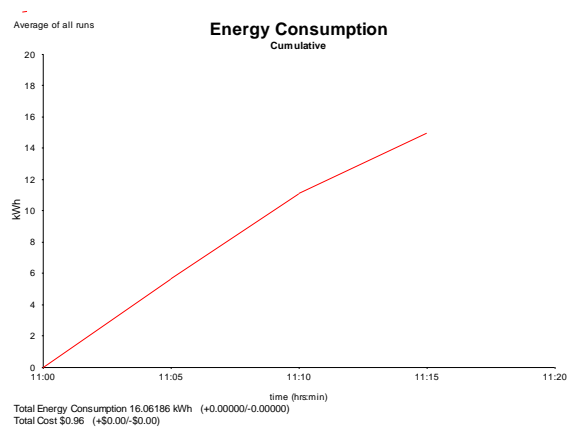
12 plantas, 5 ascensores



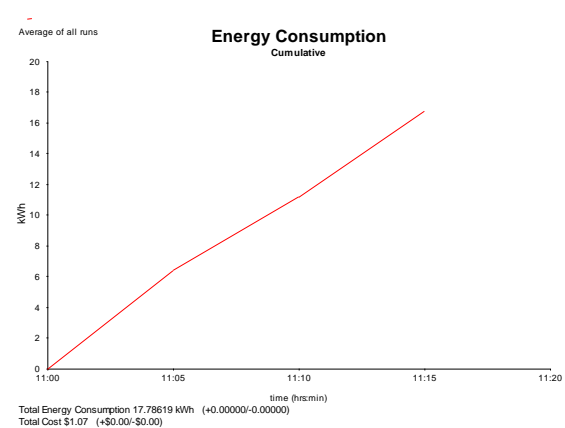
20 plantas, 8 ascensores



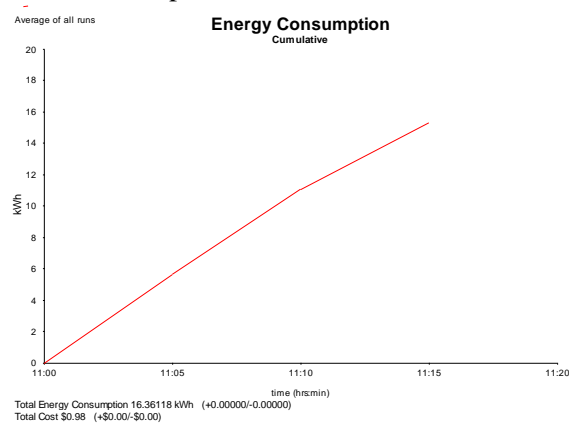
24 plantas, 10 ascensores



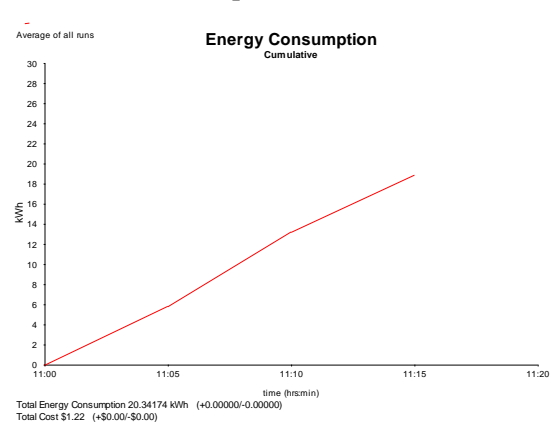
28 plantas, 13 ascensores



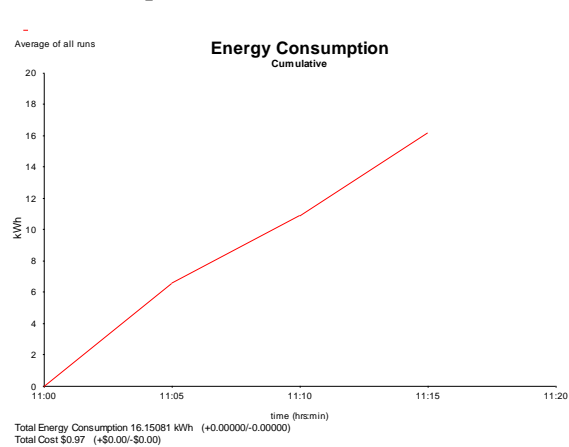
32 plantas, 14 ascensores



36 plantas, 16 ascensores



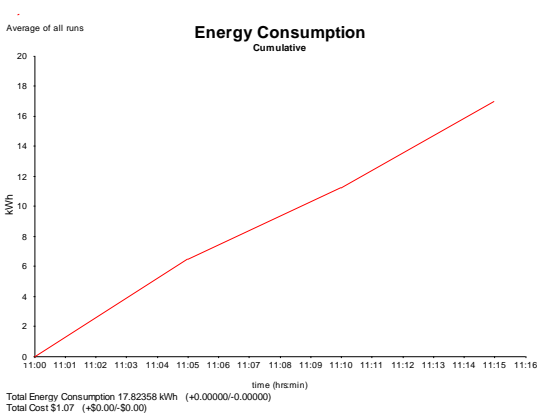
40 plantas, 19 ascensores



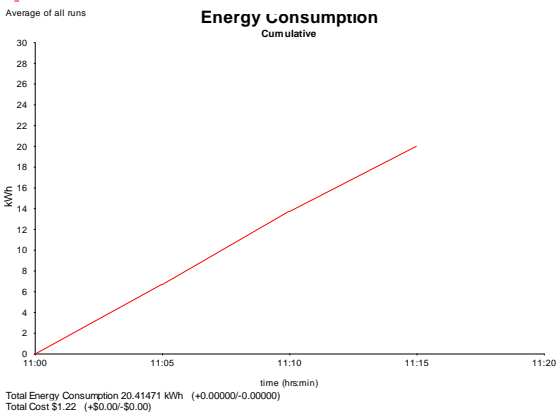
Interfloor- Función de evaluación basada en tiempo

Búsqueda Tabú

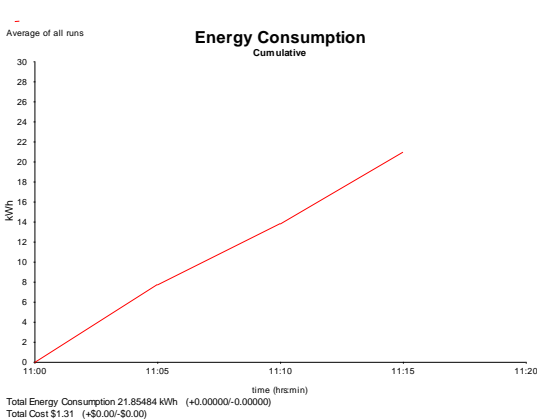
12 plantas, 5 ascensores



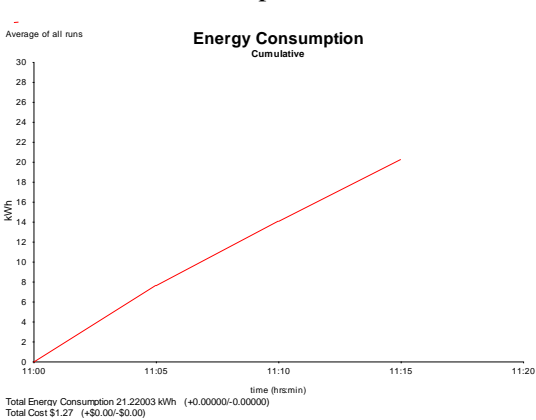
20 plantas, 8 ascensores



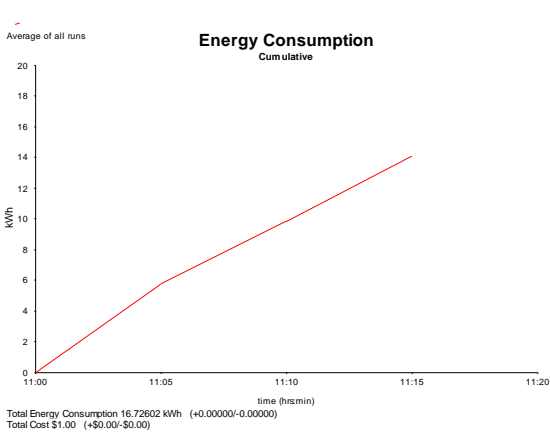
24 plantas, 10 ascensores



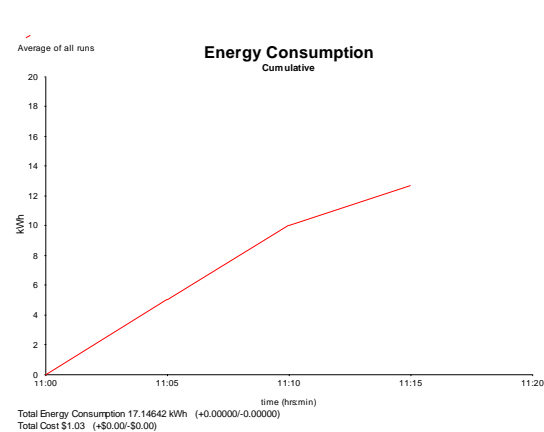
28 plantas, 13 ascensores



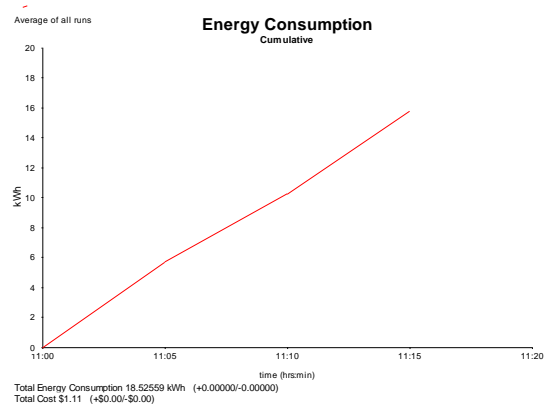
32 plantas, 14 ascensores



36 plantas, 16 ascensores

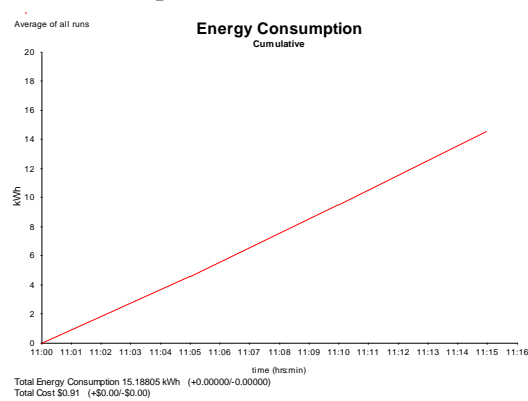


40 plantas, 19 ascensores

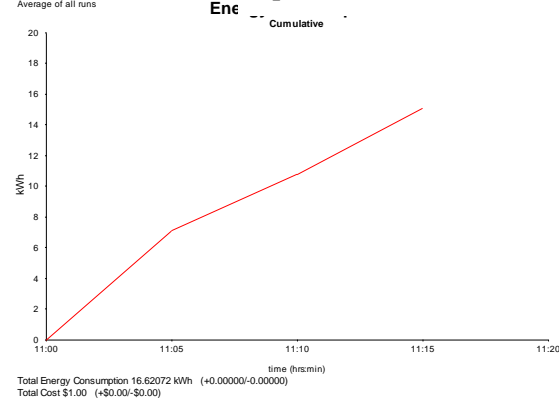


Algoritmo Genético

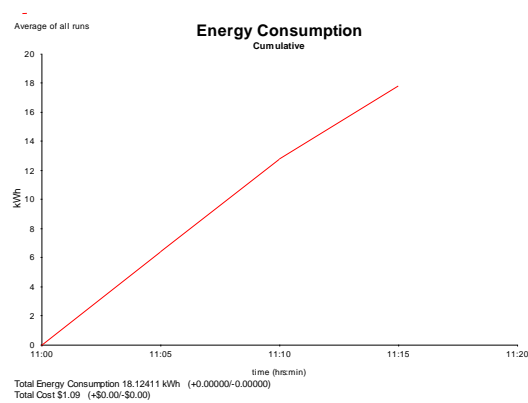
12 plantas, 5 ascensores



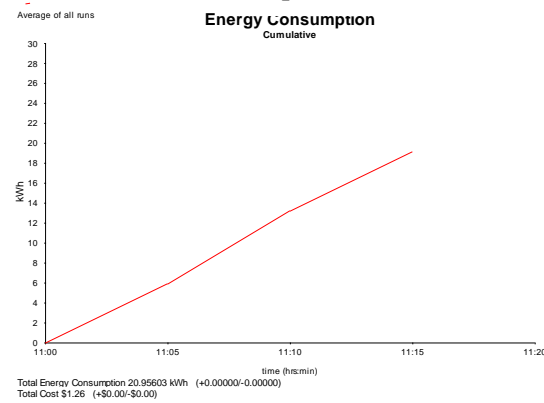
20 plantas, 8 ascensores



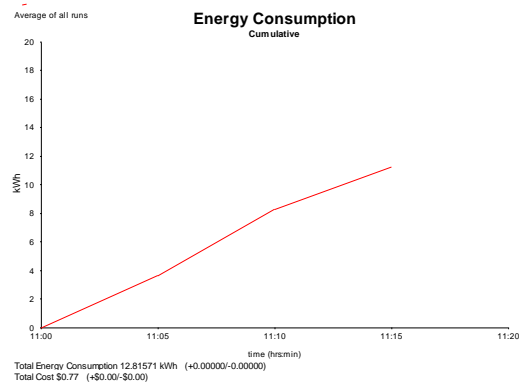
24 plantas, 10 ascensores



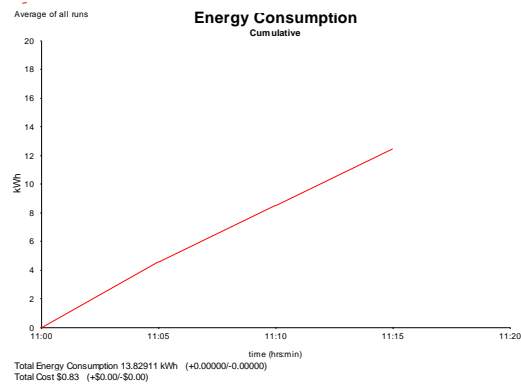
28 plantas, 13 ascensores



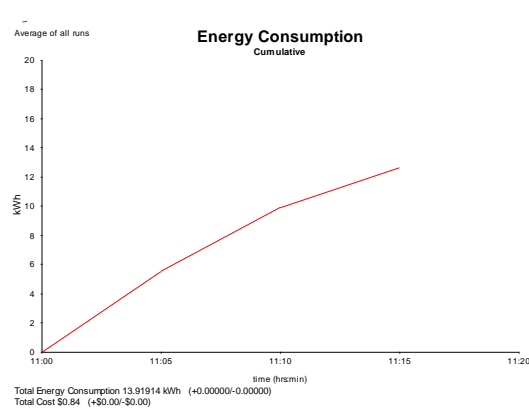
32 plantas, 14 ascensores



36 plantas, 16 ascensores

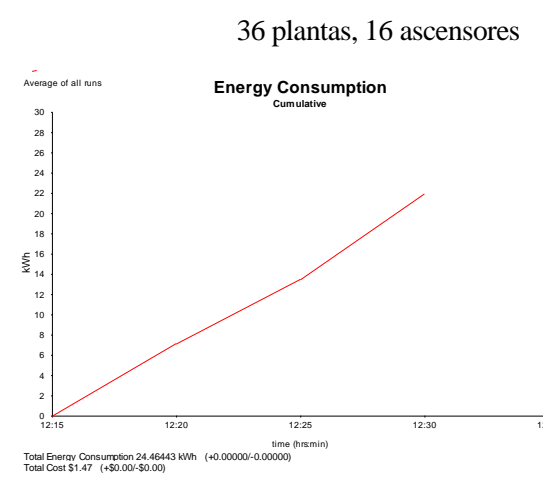
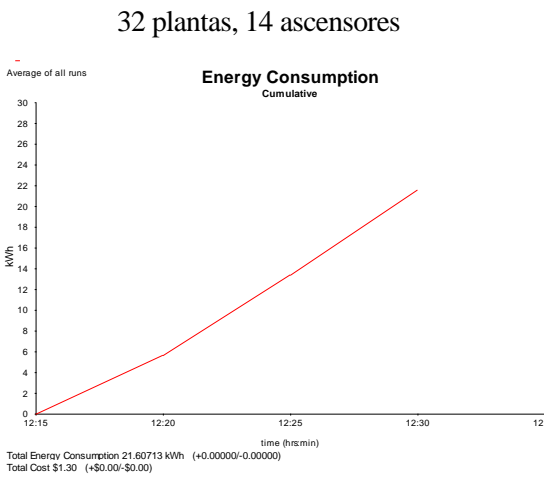
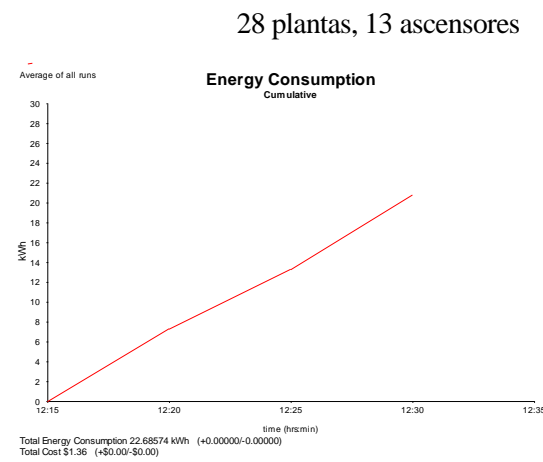
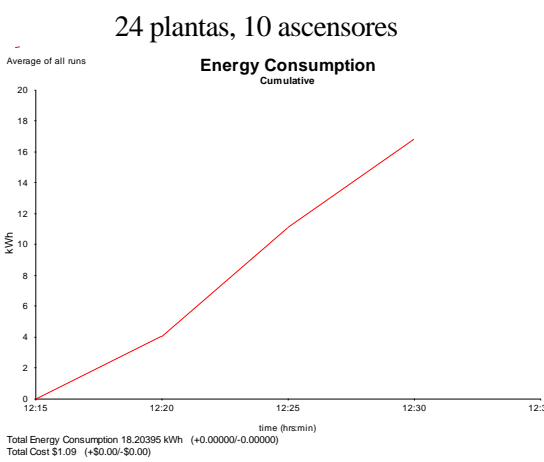
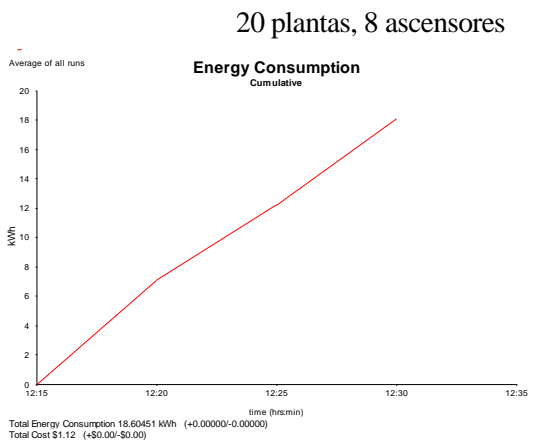
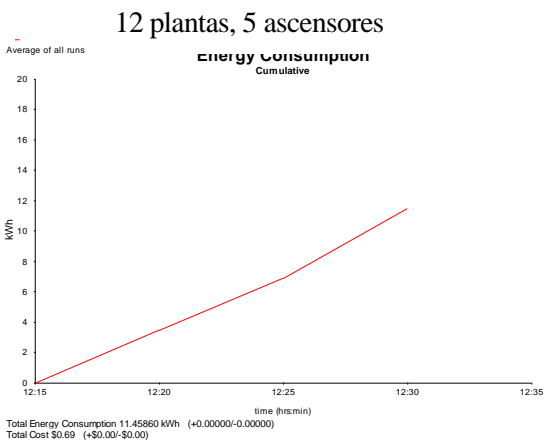


40 plantas, 19 ascensores

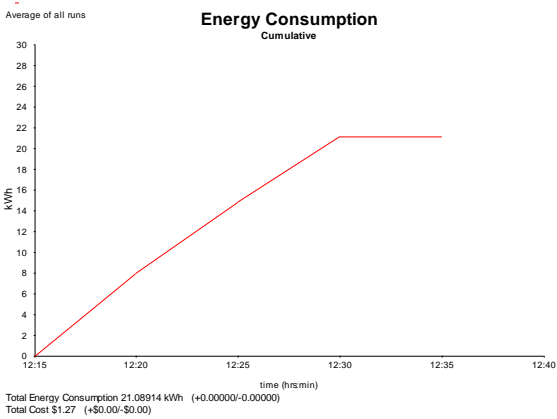


Lunchpeak- Función de evaluación basada en tiempo

Búsqueda Tabú

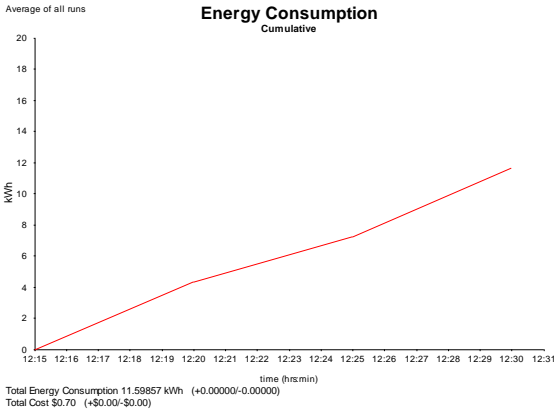


40 plantas, 19 ascensores

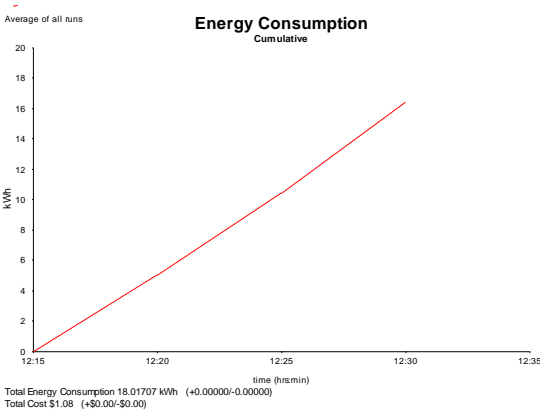


Algoritmo Genético

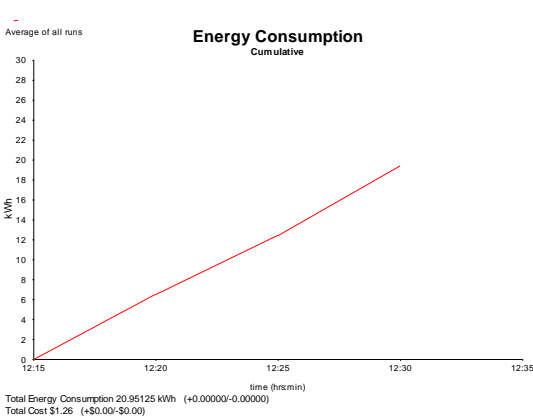
12 plantas, 5 ascensores



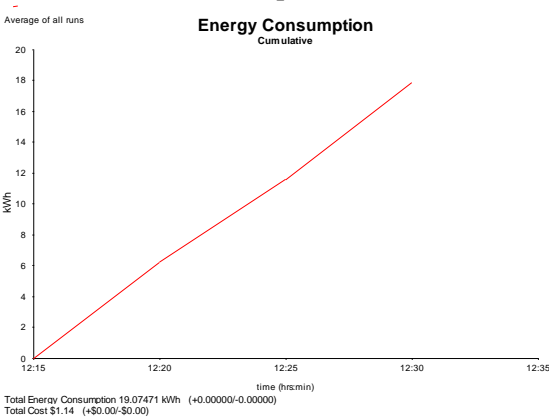
20 plantas, 8 ascensores



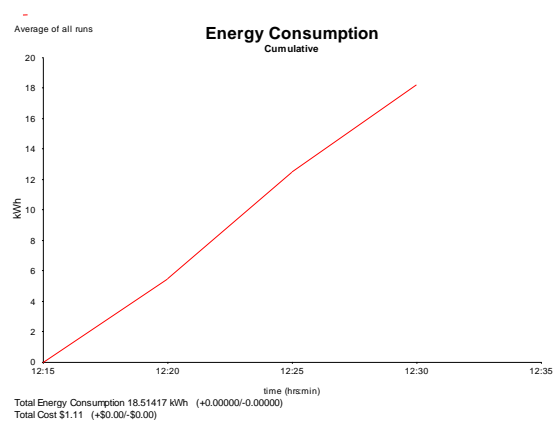
24 plantas, 10 ascensores



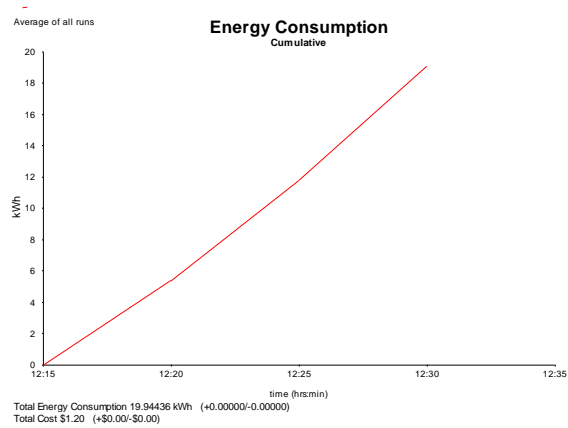
28 plantas, 13 ascensores



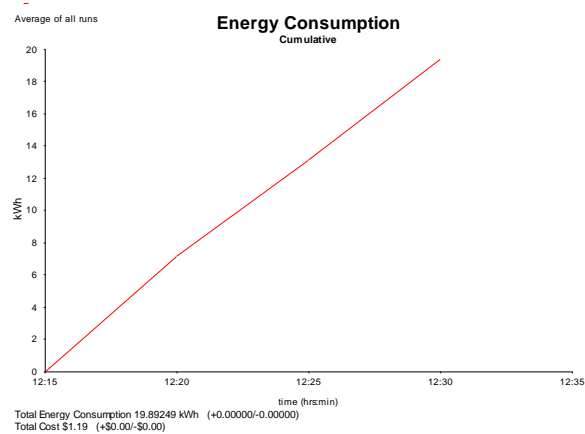
32 plantas, 14 ascensores



36 plantas, 16 ascensores



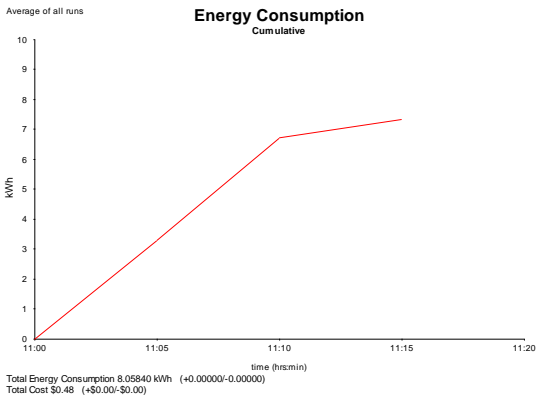
40 plantas, 19 ascensores



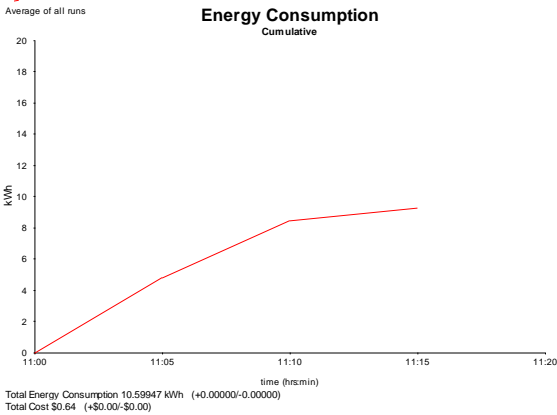
Downpeak- Función de evaluación basada en tiempo

Búsqueda Tabú

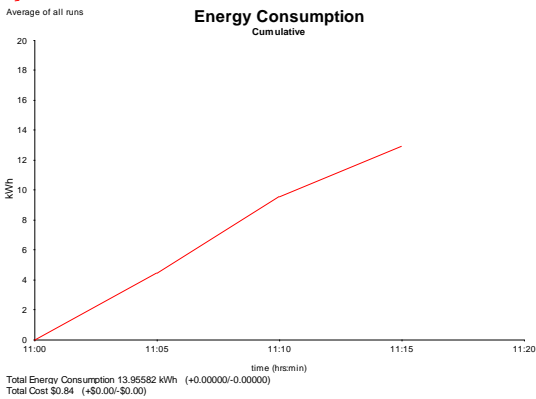
12 plantas, 5 ascensores



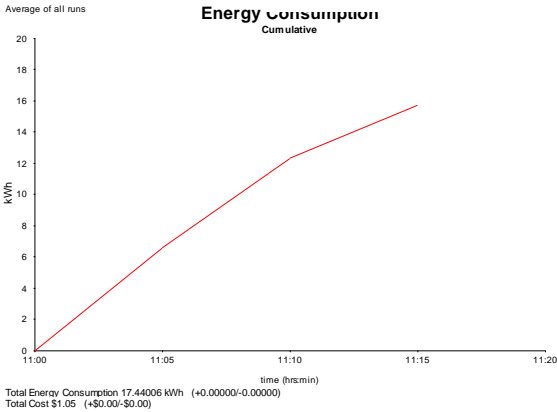
20 plantas, 8 ascensores



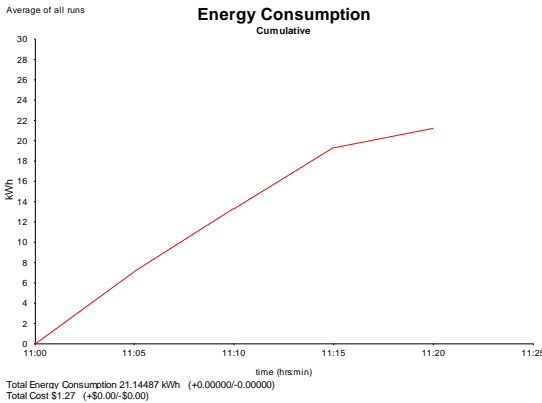
24 plantas, 10 ascensores



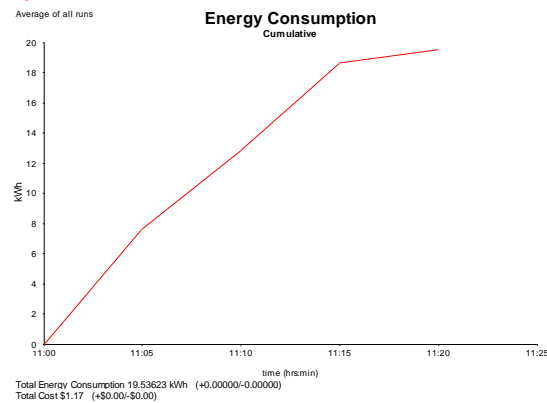
28 plantas, 13 ascensores



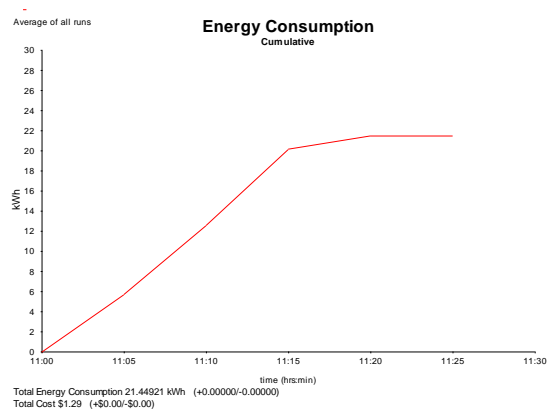
32 plantas, 14 ascensores



36 plantas, 16 ascensores

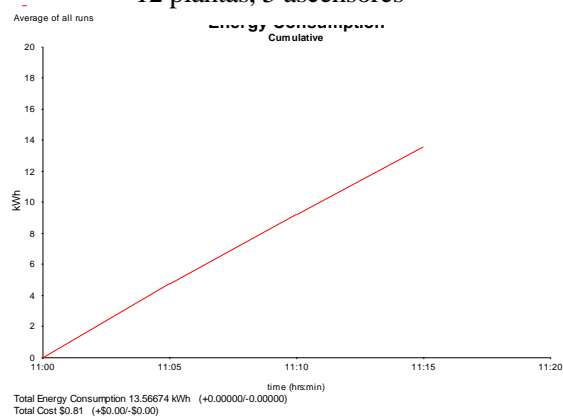


40 plantas, 19 ascensores

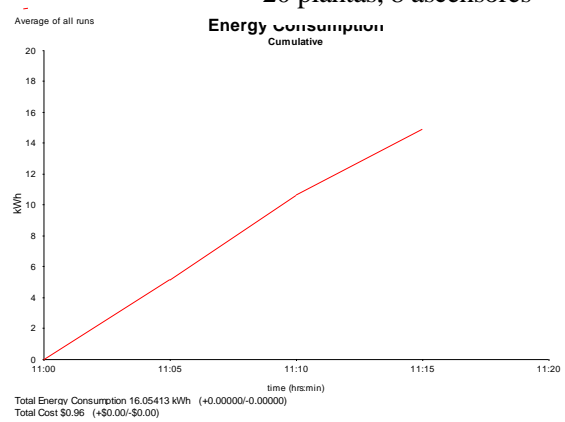


Algoritmo Genético

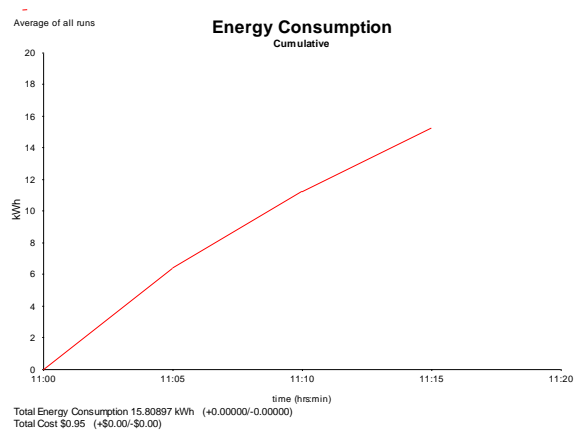
12 plantas, 5 ascensores



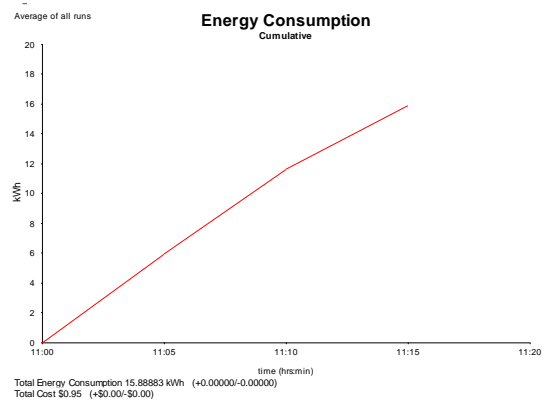
20 plantas, 8 ascensores



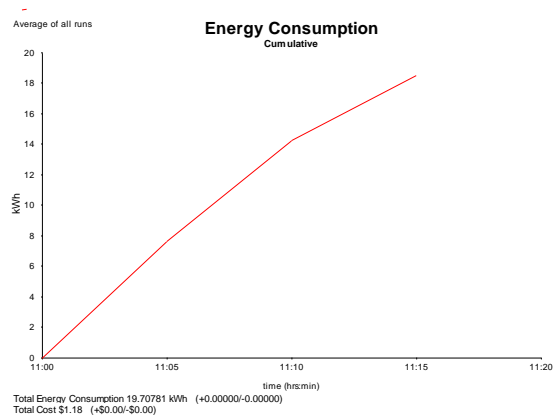
24 plantas, 10 ascensores



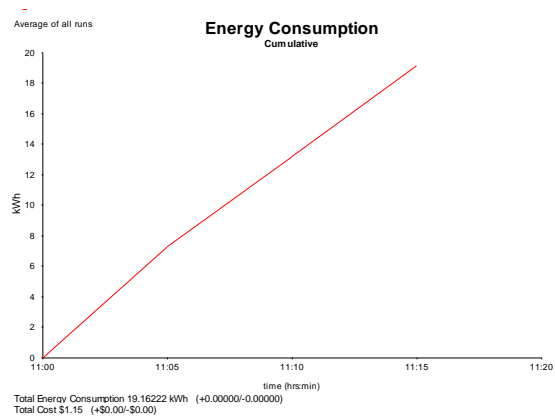
28 plantas, 13 ascensores



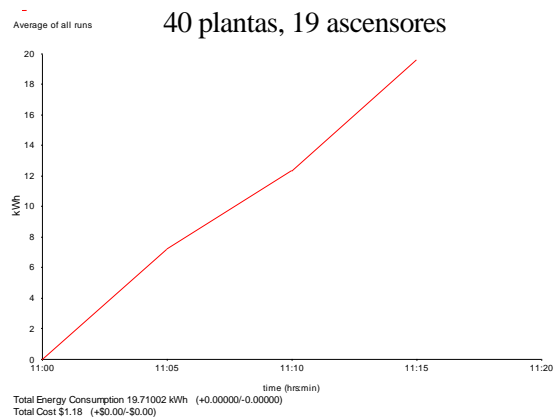
32 plantas, 14 ascensores



36 plantas, 16 ascensores



40 plantas, 19 ascensores



ANEXO C: VARIABLES DE OBJETO

La implementación de los algoritmos empleados para este estudio se ha realizado a través de una programación en C++ orientada a objetos. Debido a esto, en este Anexo C se incluyen los objetos que se han empleado para dicha programación.

Se tiene un primer objeto cuyo nombre es “*lift.h*” en el cual se encuentran las variables y características asociadas a la cabina. Entre las variables más comunes que se encuentran en este objeto se tiene el estado de las puertas del ascensor, la capacidad de carga total del ascensor, la capacidad de carga actual, la velocidad y la dirección del ascensor. Además, como se trabajan con ascensores de arquitectura *Double Deck* cada una de las variables anteriores dependen de si la cabina es superior o inferior.

El segundo objeto con el que se trabaja es el objeto “Dispatch.h” en el cual se encuentran las variables y características asociadas al edificio y a los pasajeros.

A continuación, se muestra el código de ambos objetos.

Objeto “*lift.h*”

```
/*
-----
Filename      lift.h
Copyright     (c) Peters Research Ltd
SVN variables, Updated by SVN @ each commit Id, Revision surrounded by
'$'
if the SVN file properties SVN:Keywords are set
Revision
This keyword describes the last known revision in which this file
changed in the repository
$Id: lift.h 2184 2014-10-27 15:17:59Z Jim.Nickerson $
$Revision: 2184 $
-----
*/
//
#ifdef _INC_LIFT                                //check flag to avoid re-definition
of class
#define _INC_LIFT
//
#include "stdafx.h"
//
#if _MSC_VER > 1000
#pragma once
#endif
//
#include "arraysize.h"
#include "building.h"
#include "DestinationCall.h"
//
#define MAX_DESTINATION_CALL_LIFT 1000
//
class lift
{
public:
//
enum _tag_LIFT_TYPE
{
    SINGLE_DECK = 0,           // IDSX_LIFT_TYPE_0
    DOUBLE_DECK = 1,          // IDSX_LIFT_TYPE_1
    TWO_CARS_PER_SHAFT = 2,   // IDSX_LIFT_TYPE_2
}
```

```

        TWO_D = 3,
        LAST_LIFT_TYPE // used to fill string arrays
from string table
};
//
enum _tag_CALL_CANCELLATION
{
    ARRIVAL=0,
    SLOWDOWN=1
};
//
enum _kW_STATES // the energy consumed by the drive
in each of these load conditions
    DRIVE_OFF,
    DRIVE_ON,
    DRIVE_UP_0,
    DRIVE_UP_25,
    DRIVE_UP_50,
    DRIVE_UP_75,
    DRIVE_UP_100,
    DRIVE_DOWN_0,
    DRIVE_DOWN_25,
    DRIVE_DOWN_50,
    DRIVE_DOWN_75,
    DRIVE_DOWN_100,
    DRIVE_LAST_STATE
};
//
enum _Axis
{
    AXIS_X,
    AXIS_Y,
    AXIS_Z
};
//
double m_Acceleration; //rated lift acceleration
(m/s/s) [elive static data] XMLS_1_ACCELERATION_MSS
double m_AccelerationMultiply; //not currently used, but
allows dispatcher to change the rated acceleration for a single trip.
Set to 1.0 by default.
int m_ActualQuickestStopFloor; //actual (as opposed to ideal)
quickest stop floor. Not normally used in Elevate, but when class used
in real systems
double m_ActualStoppingDistance; //actual (as opposed to ideal)
stopping distance. Not normally used in Elevate, but when class used in
real systems
int m_AlgorithmType; //algorithm type CONVENTIONAL,
DESTINATION or MIXED using global Elevate definition
bool m_Available; //normally true, can be off due
to motor generator
double m_Capacity; //nominal lift capacity
(kg) [elive static data] XMLS_1_CAPACITY_KG
int m_CarCall[MAX_FLOORS]; //car calls registered (1
registered, 0 not) [elive dynamic data] XMLS_1_REGISTERED_CALLS
int m_CarCallCancellation; //defines when lift class
cancels car calls
//options ARRIVAL
(default) or SLOWDOWN
//m_CarCall is set
when value is 1. If SLOWDOWN option selected then value is changed to 2
on slowdown
double m_CarCallDwellTime; //not used
double m_CarCallDwellTimePostPersonExit; //not used
int m_CarCallUpperCar[MAX_FLOORS]; //car calls registered (1
registered, 0 not) [elive dynamic data]

```

```

        int m_CarService; //indicates current service
state car is in (i.e. AUTOMATIC, etc) [elive dynamic data]
XMLS_1_CAR_SERVICE
        double m_CounterweightProportion; //used in conjunction with
m_VelocityMultiply to model ThyssenKrupp VMAX.
        double m_CurrentAcceleration; //current acceleration (m/s/s)
        double m_CurrentArea; //current area taken in car by
passengers (m2)
        double m_CurrentAreaUpperCar; //current area taken in upper car by
passengers (m2)
        double m_CurrentDistance; //distance travelled on current
trip (m)
        int m_CurrentFloorNo; //current floor number (where 1
is lowest floor), NONE when travelling [elive dynamic data] XMLS_1_FLOOR
        double m_CurrentJerk; //current jerk (m/s/s/s)
        double m_CurrentLoad; //current car load (kg) [elive
dynamic data] XMLS_1_CURRENT_LOADKG
        double m_CurrentLoadUpperCar; //current car load upper car (kg)
[elive dynamic data]
        double m_CurrentPosition; //current position (m above
reference)
        double m_CurrentTime; //current time (s past
reference)
        double m_CurrentVelocity; //current velocity (m/s)
        bool m_DeleteDestinationCallAtDestination; //chooses if the lift
class deletes the call when the lift arrives at destination,
//or changes state
to PENDING_RESET and saves the time the car arrives
        int DestinationByDispatcher; //Normally the lift class will decide
what call to serve next based on the
//car calls and
allocated landing calls
//set this variable
to 1 for the lift to travel to m_RequestedDestination
        int m_DestinationFloor; //current destination floor no
[elive dynamic data]
        double m_DestinationPosition; //current destination (m above
reference)
        double m_DestinationTime; //arrival time next planned
stop (s past reference)
        int m_Direction; //direction of travel (-1 down,
0 neither, 1 up) [elive dynamic data] XMLS_1_DIRECTION
        int m_DoorBeams; //flag for operation of door
beams representing passenger
//transfer (1 beams
broken, 0 clear) [elive dynamic data]
        int m_DoorBeamsRear; //flag for operation of door
beams rear doors representing passenger
//transfer (1 beams
broken, 0 clear) [elive dynamic data]
        int m_DoorBeamsUpperCar; //flag for operation of door
beams upper car representing passenger
//transfer (1 beams
broken, 0 clear) [elive dynamic data]
        int m_DoorBeamsRearUpperCar; //flag for operation of door beams
upper car rear doors representing passenger
//transfer (1 beams
broken, 0 clear) [elive dynamic data]
        double m_DoorClose; //door closing time (s)
[elive static data] XMLS_1_DOOR_CLOSE
        double m_DoorDwell1; //door dwell time 1 (s)
//corresponds to
time doors will wait until
//closing if beam
not broken

```

```

        bool m_DoorDwell1Expired;           //true if last time doors
closed, no one had entered/exited the lift
        bool m_DoorDwell1ExpiredRear; //true if last time rear doors
closed, no one had entered/exited the lift
        bool m_DoorDwell1ExpiredRearUpperCar; //true if last time upper
car rear doors closed, no one had entered/exited the lift
        bool m_DoorDwell1ExpiredUpperCar; //true if last time upper
car doors closed, no one had entered/exited the lift
        double m_DoorDwell2;                //door dwell time 2 (s)
                                           //corresponds to
time doors will wait until
                                           //closing after
beams have been broken/cleared
        int m_DoorDwellMode;                //not used
        bool m_DoorHoldOpen;                //set to true to hold doors
open, overrides m_DoorDwell2
        bool m_DoorHoldOpenRear;           //set to true to hold rear
doors open, overrides m_DoorDwell2
        double m_DoorOpen;                  //door open time (s)
[elive static data] XMLS_1_DOOR_OPEN
        double m_DoorPreOpen;               //door pre-opening (s) [elive
static data] XMLS_1_DOOR_PRE_OPEN
        double m_DoorsStart;                //time doors started
opening/closing (s past reference)
        double m_DoorsStartRear;            //time rear doors started
opening/closing (s past reference)
        double m_DoorsStartRearUpperCar; //time rear doors upper car
started opening/closing (s past reference)
        double m_DoorsStartUpperCar; //time doors upper car started
opening/closing (s past reference)
        int m_DoorStatus;                   //Door Status (1 fully open, 2
closing, 3 fully closed, 4 opening, 5 nudging) [elive dynamic data]
XMLS_1_DOOR_STATUS
        int m_DoorStatusCombined;           //used by dispatcher to
determine one value combining front and rear door status
        int m_DoorStatusCombinedUpperCar; //used by dispatcher to determine
one value combining front and rear door status upper car
        int m_DoorStatusRear;               //Door status (1 fully open, 2
closing, 3 fully closed, 4 opening, 5 nudging) [elive dynamic data]
XMLS_1_DOOR_STATUS
        int m_DoorStatusRearUpperCar; //Rear door status upper car (1 fully
open, 2 closing, 3 fully closed, 4 opening, 5 nudging) [elive dynamic
data] XMLS_1_DOOR_STATUS
        int m_DoorStatusUpperCar;           //Door status upper car (1
fully open, 2 closing, 3 fully closed, 4 opening, 5 nudging) [elive
dynamic data] XMLS_1_DOOR_STATUS
        int m_DownLandingCalls[MAX_FLOORS]; //down landing calls allocated
to lift by dispatcher (1 registered, 0 not) [elive dynamic data]
XMLS_1_DOWN_LANDING_CALL
        int m_DownLandingCallsCombined[MAX_FLOORS]; //used by dispatcher
to determine one value combining front and rear status
        int m_DownLandingCallsRear[MAX_FLOORS]; //rear down landing calls
allocated to lift by dispatcher [elive dynamic data]
        int m_DownLandingCallsRearUpperCar[MAX_FLOORS]; //rear down
landing calls upper car allocated to lift by dispatcher [elive dynamic
data]
        int m_DownLandingCallsUpperCar[MAX_FLOORS]; //down landing calls
allocated to lift by dispatcher (1 registered, 0 not) [elive dynamic
data]
        bool m_EndTravelNoCall;             //defines what happens if the
call a car is traveling to is removed
                                           //(e.g. by
dispatcher). False by default. If true, then car will stop
                                           //at next floor
(unless it has other calls to travel to).

```

```

        double m_FloorArea; //floor area of car (m²)
        double m_FloorPositions[MAX_FLOORS]; //positions of floors in
building [elive static data] // (m above
reference, can be negative) //array element [0]
not used, start with //lowest floor at
m_FloorPositions[1]
        int m_FloorsServed[MAX_FLOORS]; //indicates whether lift serves
floor [elive static data] XMLS_1_FLOORS_SERVED //used when lifts
in a group do not serve all floors //for low/high rise
groups, run separate simulations //may be changed
dynamically by dispatcher //0 not served
//1 if front doors
//2 if rear doors
//3 if front and
rear doors
        bool m_FrontDoors[MAX_FLOORS]; //true if front doors on this
landing
        int m_FrontDoorsOpenCount; //counter for m_MaxDoorReOpen
        int m_FrontDoorsOpenCountUpperCar; //counter for m_MaxDoorReOpen,
upper car
        bool m_FrontDoorsUpper[MAX_FLOORS]; //true if front doors on this
landing
        bool m_FrontLocks[MAX_FLOORS]; //true if car not allowed to
access for security reasons (even if served) [elive dynamic data]
        bool m_FrontLocksUpper[MAX_FLOORS]; //ditto for upper car [elive
dynamic data]
        bool m_FullByVision; //true if vision device says
lift if full - for use in real systems where there is a volumetric
detection device
        int m_FutureCarCalls[MAX_FLOORS]; //not used
        int m_Home; //home floor/default
parking position [elive static data] XMLS_1_HOME
        double m_HomeDoorDwell1; //alternative door dwell time
for home floor (s)
        double m_HomeDoorDwell2; //alternative door dwell time
for home floor (s)
        double m_HomePeakLandingCallDwellTime; //not used
        double m_HorizontalPosition; //horizontal position in the shaft,
e.g. [elive static data]
//1 - shaft 1
//2 - shaft 2
//1.5 - half way
between shaft 1 and 2 //user of lift
class must check that lifts do not crash!
        int m_Index; //index number of this
lift car [elive static data - allow string] XMLS_1_CARID //note that the
Elevate lift array is 1 index. 1[0] is not used or initialized.
        double m_Jerk; //rated lift jerk
(m/s/s/s) [elive static data] XMLS_1_JERK_MSSS
        double m_JourneyStart; //time lift journey started (s
past reference)
        double m_kW[ DRIVE_LAST_STATE ]; //energy consumption, this
specifies the energy consumed while in each of these load states
        int m_LandingCallCancellation; //defines when lift class
cancel hall calls

```

```

//options ARRIVAL
(default) or SLOWDOWN
//Landing calls are
set when value is 1. If SLOWDOWN option selected then value is changed
to 2 on slowdown
    double m_LandingCallDwellTime; //not used
    double m_LandingCallDwellTimePostCarCall; //not used
    double m_LevellingDelay; //levelling delay (s)
    int m_MaxDoorReOpen; //maximum number of time doors
are allowed to re-open once car calls have been registered
//set to -1 for
unlimited
    double m_MaxVelocityMultiply; //allows dispatcher to change the
rated acceleration for a single trip. Also use in ThyssenKrupp VMAX to
overspeed the lift.
    double m_MGRestartTime; //time it takes for motor
generator set to re-start (s)
    bool m_MGSet; //true if this lift has
motor generator
    double m_MGShutDownAfterTime; //time after which motor generator
set will shut down (s)
    double m_MotorStartDelay; //motor start up delay (s)
    int m_NoFloors; //no of floors in
building [elive static data] XMLS_1_NO_FLOORS
    int m_ParkCall[MAX_FLOORS]; //parking calls, like landing
call, but placed by dispatcher[elive dynamic data] XMLS_1_PARK_CALL
//lift does not
open doors on arrival
    int m_ParkOpenCall[MAX_FLOORS]; //as parking calls, but lift
parks with doors open
    bool m_PeakMode; //not used
    bool m_PersonTransferred; //not used
    int m_PreDirection; //direction of travel
once we reach current destination [elive dynamic data]
    double m_QuickestStopPosition; //next possible stop lift can
make (m above reference)
    bool m_RearDoors[MAX_FLOORS]; //true if rear doors on this landing
    int m_RearDoorsOpenCount; //not used - for future
    int m_RearDoorsOpenCountUpperCar; //not used - for future
    bool m_RearDoorsUpper[MAX_FLOORS]; //true if rear doors on this
landing
    bool m_RearLocks[MAX_FLOORS]; //true if car not allowed to access
rear doors for security reasons (even if served)[elive dynamic data]
    bool m_RearLocksUpper[MAX_FLOORS]; //true if car not allowed to
access rear doors upper car for security reasons (even if served)[elive
dynamic data]
    int m_ReasonForStopping; //not used
    bool m_RequestAvailability; //set to true to start up MG
set
    int m_RequestedDestination; //Specific destination
requested by dispatcher
    bool m_SlowDown; //true if the lift has started
slowing coming into a stop
    int m_StartFloor; //floor no current journey
started
    double m_StartPosition; //position current journey
started (m above reference)
    double m_TimeBeganStartUp; //time we requested the MG set
to be turned on
    double m_TimeLastTrip; //time last trip, used to see
if to turn off MG set
    double m_TimerT1; //time timer T1 began (s past
reference), -1 if not in use
    double m_TimerT1Rear; //time timer T1 began for rear
doors (s past reference), -1 if not in use

```



```

        double m_TimerT1RearUpperCar; //time timer T1 began for rear doors
upper car (s past reference), -1 if not in use
        double m_TimerT1UpperCar;           //time timer T1 began for upper
car (s past reference), -1 if not in use
        double m_TimerT2;                   //time timer T2 began (s past
reference), -1 if not in use
        double m_TimerT2Rear;               //time timer T2 began for rear
doors (s past reference), -1 if not in use
        double m_TimerT2RearUpperCar; //time timer T2 began for rear doors
upper car (s past reference), -1 if not in use
        double m_TimerT2UpperCar;           //time timer T2 began for upper
car (s past reference), -1 if not in use
        int m_TravelStatus;                  //current travel status,
(1 traveling, 0 at floor) [elive dynamic data] XMLS_1_TRAVEL_STATUS
        int m_Type;                         //type of lift (single,
double deck, etc.), see _tag_LIFT_TYPE
        int m_UpLandingCalls[MAX_FLOORS]; //up landing calls allocated to
lift by dispatcher (1 registered, 0 not) [elive dynamic data]
XMLS_1_UP_LANDING_CALL
        int m_UpLandingCallsCombined[MAX_FLOORS]; //used by dispatcher to
determine one value combining front and rear status
        int m_UpLandingCallsRear[MAX_FLOORS]; //rear up landing calls
[elive dynamic data]
        int m_UpLandingCallsRearUpperCar[MAX_FLOORS]; //rear up landing
calls upper car [elive dynamic data]
        int m_UpLandingCallsUpperCar[MAX_FLOORS]; //landing calls upper
car [elive dynamic data]
        double m_Velocity;                  //rated lift velocity
(m/s) [elive static data] XMLS_1_VELOCITY_MS
        double m_VelocityMultiply;          //allows dispatcher to change
the rated acceleration for a single trip. Set to 1.0 by default.
                                                //Used by
ThyssenKrupp VMAX feature.
        //
        //
        //new for multicar
        int m_HomeShaft;
        double m_VelocityXAxis;
        double m_AccelerationXAxis;
        double m_JerkXAxis;
        double m_LoadConnectionDelayXAxis;
        double m_UnLoadConnectionDelayXAxis;
        //
        int m_CurrentShaft;
        int m_NextShaft;
        int m_DirectionX;
        double m_CurrentPositionX;
        int m_DestinationShaft;
        double m_DestinationPositionX;
        double m_StartPositionX;
        double m_DestinationTimeX;
        double m_JourneyStartX;
        double m_QuickestStopPositionX;
        double m_CurrentDistanceX;
        //
        bool m_CarHold;
        //
        public:
        //constructors
        lift();
        //destructor
        ~lift() {};
        //member functions
        int ChangeJourney(int floor); //change journey, new destination,
"floor"

```

```

//use this function
having checked this is possible using QuickestFloorStopFloor();
//returns 1 if OK,
-1 if not possible to change journey
void DestinationCallUpdate(int floor, DestinationCall
m_DestCalls[MAX_DESTINATION_CALLS], building b); //updates the status of
the destination calls
int FloorAt(); //return floor no if not
traveling
int FloorNo(double position); //returns floor no at position
bool GetCarCall(int nFloor, int nDeck); //returns if there is a car
call registered for a given floor and deck
int GetDoorStatus(int nSide, int nDeck); //returns the door status
of the given deck and side
bool GetFloorsForTripServed(int nArrivalFloor, int
nDestinationFloor, int nArrivalSide, int nDestinationSide);
//returns if the
lift can serve a trip
bool GetLandingCall(int nDirection, int nFloor, int nDeck, int
nSide);
//returns true or
false depending on if there is a landing call allocated to the lift for
the specified
//lift direction,
deck and side
int GetNoDecks(); //returns the number of decks,
currently limited but related routines being developped
//for n decks for
future use
bool GetParkCall(int nFloor); //returns if there is a parking call
for the given floor
bool GetParkOpenCall(int nFloor); //returns if there is a park
open call for the given floor
int HighestFloorServed(); //Highest floor served by the
lift
int LowestFloorServed(); //Lowest floor served by the
lift
bool NoCalls(); //true if lift has no
calls at all (up, down, car, parking, etc.)
bool NoDestinationCalls(int LiftNo, DestinationCall
DestCalls[MAX_DESTINATION_CALLS]);
//true if there are
no outstanding destination calls to serve
bool OpenDoorsNewCallAtLanding(building b); //checks to see if new
call registered while lift at landing and re-opens doors if required
bool OpenDoorsNewCallAtLandingRear(); //checks to see if new call
registered rear side while lift at landing and re-opens doors if
required
bool OpenDoorsNewCallAtLandingUpperCar(); //checks to see if new
call upper car registered while lift at landing and re-opens doors if
required
bool OpenDoorsNewCallAtLandingUpperCarRear(); //checks to see if
new call registered upper car rear doors while lift at landing and re-
opens doors if required
double QuickestStopDistance(int Axis); //stopping distance if
started slowing down immediately
int QuickestFloorStopFloor(); //next stop lift could make (floor
no)
double QuickestStopPosition(); //next stop lift could make (m
above reference)
double QuickestFloorStopTime(); //time of next stop lift could
make at floor (s after midnight day 1)
double QuickestFloorStopPosition(); //next stop at floor lift
could make (position)

```

```

        double QuickestStopTime();           //time of next stop lift could
make (s after midnight day 1)
        void RemoveLandingCall(int direction, int floor); //removes
landing call - called by class when lift arrives at landing.
//May also be
called by dispatcher when re-allocating call to another lift
        void Reset(building b, int index = 0); //sets lift to home
position, cancels all calls, etc.
        void ResetDoorDwellTimers(); //Reset door dwell timers - use when
door opening complete, or if
//necessary to re-
start the door time out process (e.g. new destination
//based control
system allocation while lift is at landing, to avoid passenger missing
lift.
        void ResetDoorDwellTimersRear(); //Reset door dwell timers rear
doors
        void ResetDoorDwellTimersUpperCar(); //Reset door dwell upper car
timers
        void ResetDoorDwellTimersRearUpperCar(); //Reset door dwell timers
rear car upper lift
        void SetDestination(DestinationCall
m_DestCalls[MAX_DESTINATION_CALLS], building b); //set
destination/direction travel
        int StartJourney(int floor, building b); //start journey,
destination "floor" N.B. this sets the direction of the lift there is no
need to use SetDirection();
//returns 1 if
successful, -1 if failed e.g. because doors open
        double TimeSinceStartedJourney(); //Calculates how long lift has
been travelling on its current trip
        void Update(double CurrentTime, int Index, DestinationCall
m_DestCalls[MAX_DESTINATION_CALLS], building b);
//this function
updates the status of the lift (position, speed, door operation, etc.)
        void UpdateDestination(double CurrentTime, DestinationCall
DestCalls[MAX_DESTINATION_CALLS]);
//check for calls
allocated to lift and sets destination

        //
        private:
        void AdvancedCallCancellation();
        void EndTravelDeallocatedCalls();
        //
};
//
#endif

```

Objeto "Dispatch.h"

```
// Dispatch.h: interface for the CDispatchW class.
//
// $Id: Dispatch.h 2184 2014-10-27 15:17:59Z Jim.Nickerson $
// $Revision: 2184 $
// 12/01/2010 jimn add #if defined(DLL_USE_XML_PARAMETERS)
///////////////////////////////////////////////////////////////////

#if
!defined(AFX_DISPATCH_H__163502DF_E3E1_4DA5_BB84_9BEF27C83AB0__INCLUDED_)
)
#define AFX_DISPATCH_H__163502DF_E3E1_4DA5_BB84_9BEF27C83AB0__INCLUDED_
//
#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
//
#include "DispatchBase.h"
//
#if defined(DLL_USE_XML_PARAMETERS) // if used defined in stdafx.h, added
to return the Xml file name to Elevate
    #include "../Xml/CXml/Xml.h" // required if using xml parameters,
make sure the path is correct for your implementation
    using namespace ElevateXml; // required if using xml
parameters, this namespace is referenced
#endif
//
class CDispatchW : public CDispatchBase
{
public:
    CDispatchW();
    virtual ~CDispatchW();
    //
    // Public member functions
public:
    virtual void Update(double CurrentTime,
        lift l[MAX_LIFTS],
        double SimulationTimeStep,
        building b,
        int NoLifts,
        CString message,
        CString &mode,
        CArray<person*, person*> &PersonArray,
        int NoPassengers,
        int NoTrafficPeriods,
        double m_u[MAX_TRAFFIC_PERIODS][MAX_FLOORS],
        double
m_d[MAX_TRAFFIC_PERIODS][MAX_FLOORS][MAX_FLOORS],
        double m_StartTime[MAX_TRAFFIC_PERIODS],
        double m_EndTime[MAX_TRAFFIC_PERIODS],
        int XMLmode,
        DestinationCall DestCalls[MAX_DESTINATION_CALLS],
        CString Document,
        bool DestinationButtons[MAX_FLOORS],
        int UserSelectedMode);
```

```

//
virtual void Reset(building b, bool ResetHistory, int version);
virtual int GetNoOfAlgorithms();
virtual CString GetAlgorithmName(int nAlgorithm);
virtual int GetUpLandingCall(int floor, int side);
virtual int GetDownLandingCall(int floor, int side);
virtual void SetUpLandingCall(int floor, int val, int side);
virtual void SetDownLandingCall(int floor, int val, int side);
virtual void SetAlgorithmNo(int val);
virtual int GetAlgorithmNo();
virtual int GetAlgorithmType();
virtual CString GetDispatcherOptions();
virtual CString GetAlgorithmModeName(int nAlgorithm, int nMode);
virtual int GetNoPeakModes(int nAlgorithm);
virtual void SetDispatcherOptionsUserSelections(CString
DispatcherOptionsUserSelections);

//
// Data members
//about the dispatcher algorithms
int m_NoOfAlgorithms; //the number of
algorithms defined by the developer //in this DLL
(maximum MAX_USER_ALGORITHMS)
//
int m_DispatcherPeakMode; //peak mode dispatcher is
currently using //NORMAL, UP_PEAK,
DOWN_PEAK
//
CString m_AlgorithmName[MAX_USER_ALGORITHMS];
//the names of the algorithms - these will be //added to
drop down box in Simulation Data
//
int m_NumberOfPeakModes[MAX_USER_ALGORITHMS]; //determines
how many peak modes, e.g. normal, up peak, down peak, etc.
//
//the number of peak modes available with for algorithm
CString m_PeakModeNames[MAX_PEAK_MODES][MAX_USER_ALGORITHMS]; //not used
int m_Algorithm; //the number of the
dispatcher algorithm selected by the user

int m_AlgorithmType[MAX_USER_ALGORITHMS]; //the type of
the algorithm //0 -
conventional //1 -
destinations registered at landings
//
//about the calls registered with the dispatcher
int m_UpLandingCalls[MAX_FLOORS]; //up calls registered with the
dispatcher (1 registered, 0 not)
int m_UpLandingCallsRear[MAX_FLOORS]; //rear up landing calls
int m_DownLandingCalls[MAX_FLOORS]; //down calls registered with
the dispatcher (1 registered, 0 not)
int m_DownLandingCallsRear[MAX_FLOORS]; //rear down landing
calls

```

```

//
//The following variables provided so that you can store data
between
//calls to Update(double CurrentTime, lift l...).

int m_User1[1000];
double m_User2[1000];
bool m_RequestSimulationData; //flag to indicate that simulation
data is being requested
//by dispatcher
which is not normally available to in real life
//set this flay to
true if you want to access data from the
//person class and
the traffic arrival rate/destination matrix
//

CString m_DispatcherParameters;
CString m_DispatcherOptions;
CString m_DispatcherOptionsUserSelections;
CStdioFile m_File;

void Algorithm0(double CurrentTime, lift l[MAX_LIFTS], double
SimulationTimeStep, building b, int NoLifts,
CString message, CString &mode,
CArray<person*, person*> &PersonArray, int NoPassengers,
int NoTrafficPeriods, double
m_u[MAX_TRAFFIC_PERIODS][MAX_FLOORS],
double
m_d[MAX_TRAFFIC_PERIODS][MAX_FLOORS][MAX_FLOORS], double
m_StartTime[MAX_TRAFFIC_PERIODS],
double
m_EndTime[MAX_TRAFFIC_PERIODS], DestinationCall
DestCalls[MAX_DESTINATION_CALLS],
CString Document, bool
DestinationButtons[MAX_FLOORS], int UserSelectedMode);

void Algorithm1(double CurrentTime, lift l[MAX_LIFTS], double
SimulationTimeStep, building b, int NoLifts,
CString message, CString &mode,
CArray<person*, person*> &PersonArray, int NoPassengers,
int NoTrafficPeriods, double
m_u[MAX_TRAFFIC_PERIODS][MAX_FLOORS],
double
m_d[MAX_TRAFFIC_PERIODS][MAX_FLOORS][MAX_FLOORS], double
m_StartTime[MAX_TRAFFIC_PERIODS],
double
m_EndTime[MAX_TRAFFIC_PERIODS], DestinationCall
DestCalls[MAX_DESTINATION_CALLS],
CString Document, bool
DestinationButtons[MAX_FLOORS], int UserSelectedMode);

void Algorithm2(double CurrentTime, lift l[MAX_LIFTS], double
SimulationTimeStep, building b, int NoLifts,
CString message, CString &mode,
CArray<person*, person*> &PersonArray, int NoPassengers,
int NoTrafficPeriods, double
m_u[MAX_TRAFFIC_PERIODS][MAX_FLOORS],
double
m_d[MAX_TRAFFIC_PERIODS][MAX_FLOORS][MAX_FLOORS], double
m_StartTime[MAX_TRAFFIC_PERIODS],
double
m_EndTime[MAX_TRAFFIC_PERIODS], DestinationCall
DestCalls[MAX_DESTINATION_CALLS],

```

```

        CString Document, bool
DestinationButtons[MAX_FLOORS], int UserSelectedMode);

    void Algorithm3(double CurrentTime, lift l[MAX_LIFTS], double
SimulationTimeStep, building b, int NoLifts,
        CString message, CString &mode,
CArray<person*, person*> &PersonArray, int NoPassengers,
        int NoTrafficPeriods, double
m_u[MAX_TRAFFIC_PERIODS][MAX_FLOORS],
        double
m_d[MAX_TRAFFIC_PERIODS][MAX_FLOORS][MAX_FLOORS], double
m_StartTime[MAX_TRAFFIC_PERIODS],
        double
m_EndTime[MAX_TRAFFIC_PERIODS], DestinationCall
DestCalls[MAX_DESTINATION_CALLS],
        CString Document, bool
DestinationButtons[MAX_FLOORS], int UserSelectedMode);

    void Algorithm4(double CurrentTime, lift l[MAX_LIFTS], double
SimulationTimeStep, building b, int NoLifts,
        CString message, CString &mode,
CArray<person*, person*> &PersonArray, int NoPassengers,
        int NoTrafficPeriods, double
m_u[MAX_TRAFFIC_PERIODS][MAX_FLOORS],
        double
m_d[MAX_TRAFFIC_PERIODS][MAX_FLOORS][MAX_FLOORS], double
m_StartTime[MAX_TRAFFIC_PERIODS],
        double
m_EndTime[MAX_TRAFFIC_PERIODS], DestinationCall
DestCalls[MAX_DESTINATION_CALLS],
        CString Document, bool
DestinationButtons[MAX_FLOORS], int UserSelectedMode);

    void Algorithm5(double CurrentTime, lift l[MAX_LIFTS], double
SimulationTimeStep, building b, int NoLifts,
        CString message, CString &mode,
CArray<person*, person*> &PersonArray, int NoPassengers,
        int NoTrafficPeriods, double
m_u[MAX_TRAFFIC_PERIODS][MAX_FLOORS],
        double
m_d[MAX_TRAFFIC_PERIODS][MAX_FLOORS][MAX_FLOORS], double
m_StartTime[MAX_TRAFFIC_PERIODS],
        double
m_EndTime[MAX_TRAFFIC_PERIODS], DestinationCall
DestCalls[MAX_DESTINATION_CALLS],
        CString Document, bool
DestinationButtons[MAX_FLOORS], int UserSelectedMode);

    void Algorithm6(double CurrentTime, lift l[MAX_LIFTS], double
SimulationTimeStep, building b, int NoLifts,
        CString message, CString &mode,
CArray<person*, person*> &PersonArray, int NoPassengers,
        int NoTrafficPeriods, double
m_u[MAX_TRAFFIC_PERIODS][MAX_FLOORS],
        double
m_d[MAX_TRAFFIC_PERIODS][MAX_FLOORS][MAX_FLOORS], double
m_StartTime[MAX_TRAFFIC_PERIODS],
        double
m_EndTime[MAX_TRAFFIC_PERIODS], DestinationCall
DestCalls[MAX_DESTINATION_CALLS],
        CString Document, bool
DestinationButtons[MAX_FLOORS], int UserSelectedMode);

```

```

    void Algorithm7(double CurrentTime, lift l[MAX_LIFTS], double
SimulationTimeStep, building b, int NoLifts,
                                CString message, CString &mode,
CArray<person*, person*> &PersonArray, int NoPassengers,
                                int NoTrafficPeriods, double
m_u[MAX_TRAFFIC_PERIODS][MAX_FLOORS],
                                double
m_d[MAX_TRAFFIC_PERIODS][MAX_FLOORS][MAX_FLOORS], double
m_StartTime[MAX_TRAFFIC_PERIODS],
                                double
m_EndTime[MAX_TRAFFIC_PERIODS], DestinationCall
DestCalls[MAX_DESTINATION_CALLS],
                                CString Document, bool
DestinationButtons[MAX_FLOORS], int UserSelectedMode);

    void Algorithm8(double CurrentTime, lift l[MAX_LIFTS], double
SimulationTimeStep, building b, int NoLifts,
                                CString message, CString &mode,
CArray<person*, person*> &PersonArray, int NoPassengers,
                                int NoTrafficPeriods, double
m_u[MAX_TRAFFIC_PERIODS][MAX_FLOORS],
                                double
m_d[MAX_TRAFFIC_PERIODS][MAX_FLOORS][MAX_FLOORS], double
m_StartTime[MAX_TRAFFIC_PERIODS],
                                double
m_EndTime[MAX_TRAFFIC_PERIODS], DestinationCall
DestCalls[MAX_DESTINATION_CALLS],
                                CString Document, bool
DestinationButtons[MAX_FLOORS], int UserSelectedMode);

    void Algorithm9(double CurrentTime, lift l[MAX_LIFTS], double
SimulationTimeStep, building b, int NoLifts,
                                CString message, CString &mode,
CArray<person*, person*> &PersonArray, int NoPassengers,
                                int NoTrafficPeriods, double
m_u[MAX_TRAFFIC_PERIODS][MAX_FLOORS],
                                double
m_d[MAX_TRAFFIC_PERIODS][MAX_FLOORS][MAX_FLOORS], double
m_StartTime[MAX_TRAFFIC_PERIODS],
                                double
m_EndTime[MAX_TRAFFIC_PERIODS], DestinationCall
DestCalls[MAX_DESTINATION_CALLS],
                                CString Document, bool
DestinationButtons[MAX_FLOORS], int UserSelectedMode);

    void Algorithm10(double CurrentTime, lift l[MAX_LIFTS], double
SimulationTimeStep, building b, int NoLifts,
                                CString message, CString &mode,
CArray<person*, person*> &PersonArray, int NoPassengers,
                                int NoTrafficPeriods, double
m_u[MAX_TRAFFIC_PERIODS][MAX_FLOORS],
                                double
m_d[MAX_TRAFFIC_PERIODS][MAX_FLOORS][MAX_FLOORS], double
m_StartTime[MAX_TRAFFIC_PERIODS],
                                double
m_EndTime[MAX_TRAFFIC_PERIODS], DestinationCall
DestCalls[MAX_DESTINATION_CALLS],
                                CString Document, bool
DestinationButtons[MAX_FLOORS], int UserSelectedMode);

```



```

    void Algorithm11(double CurrentTime, lift l[MAX_LIFTS], double
SimulationTimeStep, building b, int NoLifts,
                                CString message, CString &mode,
CArray<person*, person*> &PersonArray, int NoPassengers,
                                int NoTrafficPeriods, double
m_u[MAX_TRAFFIC_PERIODS][MAX_FLOORS],
                                double
m_d[MAX_TRAFFIC_PERIODS][MAX_FLOORS][MAX_FLOORS], double
m_StartTime[MAX_TRAFFIC_PERIODS],
                                double
m_EndTime[MAX_TRAFFIC_PERIODS], DestinationCall
DestCalls[MAX_DESTINATION_CALLS],
                                CString Document, bool
DestinationButtons[MAX_FLOORS], int UserSelectedMode);

    bool GetDispatcherOptionsValueBool(CString VariableName);
    double GetDispatcherOptionsValueNumber(CString VariableName);
    double GetDispatcherOptionsValueTime(CString VariableName);
    CString GetDispatcherOptionsValueCombo(CString VariableName);
    void SetValue(CString VariableName, double VariableValue);
    void SetValue(CString VariableName, int index1, double
VariableValue);
    void SetValue(CString VariableName, int index1, int index2, double
VariableValue);
    double GetValue(CString VariableName);
    double GetValue(CString VariableName, int index1);
    double GetValue(CString VariableName, int index1, int index2);
    CString NumberText(double val);
    CString NumberText(int val);
#ifdef DLL_USE_XML_PARAMETERS
    void SetDispatcherDllOptions( CXmlNode DllOptions, CString
csAlgorithmName ); // if supports Xml Parameters
    void GetOptionsFromXml( CString csAlgorithmName ); // extract the
Option Value names and values from the Xml file for this algorithm
#endif
protected:
};
#endif //
!defined(AFX_DISPATCH_H__163502DF_E3E1_4DA5_BB84_9BEF27C83AB0__INCLUDED_
)

```